

Las prácticas en laboratorios forman una parte muy importante de la formación en todos los programas docentes. A pesar de esta importancia, la creación de un laboratorio no es una tarea fácil, ya que el hecho de equipar un laboratorio puede suponer un gran gasto económico, tanto inicial como posterior.

Como solución, surge la educación a distancia, y en concreto los laboratorios virtuales, es decir, simulaciones de un laboratorio real utilizando modelos matemáticos. Por sus características y flexibilidad se han ido desarrollando laboratorios virtuales en el ámbito docente, pero no todas las áreas cuentan con tantas posibilidades o facilidades como en la electrónica.

La mayoría de los laboratorios accesibles desde Internet que hay en la actualidad dentro de la enseñanza a distancia o formación online, son virtuales. El laboratorio que se ha desarrollado tiene como principal ventaja la realización de prácticas controlando instrumentos y circuitos reales de forma remota.

El proyecto consiste en realizar un sistema software para implementar un laboratorio remoto en el área de la electrónica analógica, que pueda ser utilizado como complemento a las actividades formativas que se realizan en los laboratorios de los centros de enseñanza.

El sistema completo también consta de un hardware controlado mediante buses de comunicación estándar, que permite la implementación de distintos circuitos analógicos, de tal forma que se pueda realizar prácticas sobre circuitos físicos reales.

Para desarrollar un laboratorio lo más real posible, la aplicación que maneja el estudiante es un **visor 3D**.

Con la utilización de un visor 3D lo que se pretende es tener un aumento de la realidad a la hora de realizar las prácticas de laboratorio remotamente.

El sistema desarrollado cuenta con un sistema de comunicación basado en un modelo cliente-servidor:

- **Servidor:** se encarga de procesar las acciones que realiza el cliente y controla y monitoriza los instrumentos y dispositivos del sistema hardware.
- **Cliente:** sería el usuario final, que mediante un visor 3D comunica las acciones a realizar al servidor para que éste las procese.

Practices in laboratories are a very important part of training in all educational programs. Despite this importance, the establishment of a laboratory is not an easy task, since the fact of equipping a laboratory can be a great economic budget, both initial and subsequent spending.

As a solution, appears the education at distance (online), and in particular the virtual labs, namely simulations of a real laboratory by using mathematical models. Virtual laboratories in the field of teaching have been developed for its features and flexibility, but not all areas have so many possibilities or facilities as in electronics.

The most accessible laboratories from the Internet that are currently accessible within the distance or e-learning (on-line) are virtual. The laboratory which has been developed has as a main advantage to make practices or exercises in the fact of controlling instruments and real circuits remotely.

The project consists of making a software system in order to implement a remote laboratory in the area of analog electronics that can be used as a complement to the others training activities to be carried out.

The complete system also consists of a controlled hardware by standard communication buses that allow the implementation of several analog circuits, in such a way that practices can control real physical circuits.

To develop a laboratory as more realistic as possible, the application that manages the student is a 3D viewer.

With the use of a 3D viewer, is intended to have an increase in reality when any student wants to access to laboratory practices remotely.

The developed system has a communication system based on a model Client/Server:

- **Server:** The system that handles actions provided by the client and controls and monitors the instruments and devices in the hardware system.
- **Client:** The end user, which using a 3D viewer, communicates the actions to be performed at the server so that it will process them.



# **UNIVERSIDAD POLITÉCNICA DE MADRID**

## **ESCUELA UNIVERSITARIA DE INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN**

### **PROYECTO FIN DE CARRERA**

**“DESARROLLO DE UNA APLICACIÓN SOFTWARE PARA  
LABORATORIOS REMOTOS”**

**Alumno:** Daniel Contreras Masedo

**Tutor:** Sergio López Gregorio

**Sistemas Electrónicos  
Madrid, Septiembre 2012**





# *AGRADECIMIENTOS*

En primer lugar, quiero agradecer a mis padres, Antonio y Marisa, por darme la oportunidad de estudiar esta carrera y ofrecerme el apoyo y la confianza necesaria durante estos años, que me ha ayudado a continuar y no rendirme.

A mis hermanos, Miguel y Darío, y mi prima Nuria, por haber estado ahí siempre que os he necesitado.

A ti Ángela, por estar a mi lado, aguantarme y hacerme sonreír cada día. No cambies nunca.

También me gustaría agradecer a toda la gente, amigos y compañeros, los momentos que he vivido gracias a ellos, y que han hecho un poquito más fácil recorrer este camino.

Por último y no menos importante, quiero agradecer a mi tutor, Sergio López, la confianza depositada en mí. Que en un futuro (no muy lejano), podamos decir que nosotros formamos parte de los laboratorios que utilicen los futuros estudiantes.

En definitiva, ¡GRACIAS! a todos por haber estado conmigo en esta etapa de mi vida (que por fin termina).



# INDICE

|   |            |
|---|------------|
| INDICE DE FIGURAS .....   | 3          |
| INDICE DE TABLAS.....   | 7          |
| <b>1 INTRODUCCIÓN, DESCRIPCIÓN Y OBJETIVOS .....</b>                        | <b>9</b>   |
| 1.1 <i>Introducción</i> .....   | 11         |
| 1.2 <i>Objetivos</i> .....  | 12         |
| 1.3 <i>Descripción del proyecto</i> .....                                   | 13         |
| <b>2 LABORATORIO REMOTO.....</b>  | <b>15</b>  |
| 2.1 <i>Introducción</i> .....   | 17         |
| 2.2 <i>Tipos de laboratorios</i> .....                                      | 17         |
| 2.3 <i>Tipologías de laboratorios remotos</i> .....                         | 22         |
| 2.4 <i>Análisis de requisitos para la implementación</i> .....              | 23         |
| 2.5 <i>Estructura y componentes de un laboratorio remoto</i> .....          | 24         |
| 2.6 <i>Ejemplos de laboratorios remotos</i> .....                           | 26         |
| <b>3 SOLUCIÓN ESCOGIDA PARA EL LABORATORIO REMOTO .....</b>                 | <b>39</b>  |
| 3.1 <i>Introducción</i> .....   | 41         |
| 3.2 <i>Especificaciones generales</i> .....                                 | 42         |
| 3.3 <i>Sistema hardware</i> .....   | 43         |
| 3.4 <i>Aplicación software</i> .....  | 46         |
| 3.5 <i>Sistema final: Laboratorio remoto de electrónica analógica</i> ..... | 51         |
| <b>4 DESARROLLO DEL SISTEMA SOFTWARE .....</b>                              | <b>53</b>  |
| 4.1 <i>Introducción</i> .....   | 55         |
| 4.2 <i>Aplicación cliente: Visor 3D</i> .....                               | 56         |
| 4.2.1 <i>Entorno virtual</i> .....  | 57         |
| 4.2.2 <i>Listado de visores</i> .....                                       | 62         |
| 4.2.3 <i>Requisitos para la instalación del visor</i> .....                 | 64         |
| 4.3 <i>Sistema software del servidor</i> .....                              | 65         |
| 4.3.1 <i>Opensim</i> .....  | 65         |
| 4.3.2 <i>Aplicación servidor</i> .....                                      | 88         |
| 4.4 <i>Visión global del sistema software</i> .....                         | 113        |
| <b>5 MANUAL DE USUARIO I: VISOR 3D .....</b>                                | <b>115</b> |
| 5.1 <i>Introducción</i> .....   | 117        |
| 5.2 <i>Visor 3D</i> .....   | 118        |
| 5.2.1 <i>Instalación</i> .....  | 118        |
| 5.2.2 <i>Manejo básico del visor</i> .....                                  | 120        |
| 5.2.3 <i>Otras opciones del visor</i> .....                                 | 128        |

|       |  |     |
|-------|--|-----|
| 5.3   | <i>Tutorial de uso del laboratorio eLab-3D</i> ..... | 128 |
| 5.3.1 | <i>Isla</i> .....                                    | 129 |
| 5.3.2 | <i>Parque de esparcimiento</i> .....                 | 130 |
| 5.3.3 | <i>Edificio</i> .....                                | 130 |
| 5.3.4 | <i>Puesto de laboratorio</i> .....                   | 135 |
| 5.3.5 | <i>Pruebas libres</i> .....                          | 175 |
| 6     | MANUAL DE USUARIO II: SERVIDOR LABORATORIO .....     | 177 |
| 6.1   | <i>Introducción</i> .....                            | 179 |
| 7     | PRESUPUESTO.....                                     | 185 |
| 7.1   | <i>Software</i> .....                                | 187 |
| 7.2   | <i>Hardware</i> .....                                | 187 |
| 7.3   | <i>Mano de obra</i> .....                            | 188 |
| 7.4   | <i>Total</i> .....                                   | 189 |
| 8     | CONCLUSIONES Y TRABAJO FUTURO.....                   | 191 |
| 8.1   | <i>Conclusiones</i> .....                            | 193 |
| 8.2   | <i>Trabajo futuro</i> .....                          | 194 |
| 9     | ANEXO I .....  | 195 |
| 9.1   | <i>Instalación Phoenix Viewer</i> .....              | 197 |
| 9.2   | <i>Instalación SecondLife Viewer</i> .....           | 199 |
| 10    | REFERENCIAS.....                                     | 201 |
| 10.1  | <i>Documentos</i> .....                              | 203 |
| 10.2  | <i>Consultas on-line</i> .....                       | 204 |

# INDICE DE FIGURAS

|  |    |
|--|----|
| Figura 1.3-1: Sistema software del laboratorio remoto.....                 | 14 |
| Figura 2.2-1: Comparativa entre L.Tradicional y L.Remoto.....              | 19 |
| Figura 2.5-1: Componentes típicos de un laboratorio remoto .....           | 25 |
| Figura 2.6-1: Web Laboratorio WebLab Deusto .....                          | 27 |
| Figura 2.6-2: Mesa de trabajo tradicional y on-line .....                  | 29 |
| Figura 2.6-3: Placa virtual del laboratorio remoto VISIR.....              | 29 |
| Figura 2.6-4: Implementación del laboratorio remoto VISIR.....             | 30 |
| Figura 2.6-5: Osciloscopio del laboratorio remoto VISIR .....              | 31 |
| Figura 2.6-6: Sistema hardware empleado en laboratorio remoto VISIR .....  | 31 |
| Figura 2.6-7: Estructura global NetLab.....                                | 32 |
| Figura 2.6-8: Ejemplo laboratorio remoto NetLab.....                       | 33 |
| Figura 2.6-9: Equipamiento real laboratorio remoto NetLab .....            | 34 |
| Figura 2.6-10: Ejemplo de utilización del laboratorio remoto ISILab.....   | 35 |
| Figura 2.6-11: hardware donde se integran los circuitos en el ISILab ..... | 36 |
| Figura 3.3-1: Sistema hardware del laboratorio remoto .....                | 43 |
| Figura 3.3-2: Esquema general de la placa Análisis de Circuitos .....      | 44 |
| Figura 3.3-3: Esquema del divisor de tensión .....                         | 45 |
| Figura 3.4-1: Sistema software del servidor.....                           | 48 |
| Figura 3.5-1: Sistema completo del servidor .....                          | 52 |
| Figura 4.1-1: Sistema software completo .....                              | 55 |
| Figura 4.2-1: Visor del entorno 3D .....                                   | 56 |
| Figura 4.3-1: Logo Opensim .....   | 65 |
| Figura 4.3-2: Consola de comandos Opensim.....                             | 67 |
| Figura 4.3-3: Arquitectura Standalone.....                                 | 69 |
| Figura 4.3-4: Arquitectura Grid .....                                      | 70 |
| Figura 4.3-5: Isla inicial del mundo virtual .....                         | 71 |
| Figura 4.3-6: Ejemplo de isla ya construida .....                          | 72 |
| Figura 4.3-7: Ventana de construcción .....                                | 73 |
| Figura 4.3-8: Estructura del lenguaje LSL .....                            | 75 |
| Figura 4.3-9: Ejemplo de un script en LSL .....                            | 77 |
| Figura 4.3-10: Objeto en mundo virtual con su script .....                 | 78 |
| Figura 4.3-11: Código de ejemplo de la función llSay .....                 | 79 |
| Figura 4.3-12: Código de ejemplo de la función llMessageLinked.....        | 80 |
| Figura 4.3-13: Registros de comunicaciones.....                            | 82 |
| Figura 4.3-14: Comunicación Opensim-Servicio Web.....                      | 84 |
| Figura 4.3-15: Ejemplo de script con función llHTTPRequest .....           | 84 |
| Figura 4.3-16: Ejemplo comunicación mediante http_request .....            | 85 |
| Figura 4.3-17: Edificio Laboratorio Electrónica .....                      | 86 |
| Figura 4.3-18: Entrada del laboratorio de electrónica .....                | 87 |
| Figura 4.3-19: Puesto de laboratorio .....                                 | 87 |
| Figura 4.3-20: Partes del servidor de laboratorio .....                    | 88 |

|  |     |
|--|-----|
| Figura 4.3-21: Panel frontal y diagrama de bloques de un programa de LabView ..... | 89  |
| Figura 4.3-22: Llamada a un servicio web .....                                     | 90  |
| Figura 4.3-23: Figura 1 manual Servicio Web.....                                   | 91  |
| Figura 4.3-24: Figura 2 manual Servicio Web.....                                   | 92  |
| Figura 4.3-25: Figura 3 manual Servicio Web.....                                   | 93  |
| Figura 4.3-26: Figura 4 manual Servicio Web.....                                   | 93  |
| Figura 4.3-27: Figura 5 manual Servicio Web.....                                   | 94  |
| Figura 4.3-28: Figura 6 manual Servicio Web.....                                   | 95  |
| Figura 4.3-29: Figura 7 manual Servicio Web.....                                   | 96  |
| Figura 4.3-30: Figura 8 manual Servicio Web.....                                   | 96  |
| Figura 4.3-31: Figura 9 manual Servicio Web.....                                   | 97  |
| Figura 4.3-32: Figura 10 manual Servicio Web.....                                  | 98  |
| Figura 4.3-33: Figura 11 manual Servicio Web.....                                  | 98  |
| Figura 4.3-34: Figura 12 manual Servicio Web.....                                  | 99  |
| Figura 4.3-35: Ejemplo de URL .....  | 100 |
| Figura 4.3-36: Comparativa URL de acceso con el VI correspondiente.....            | 100 |
| Figura 4.3-37: VI servicio web del generador de funciones .....                    | 102 |
| Figura 4.3-38: Partes del VI del servicio web.....                                 | 102 |
| Figura 4.3-39: Flujograma del Servidor Laboratorio.....                            | 104 |
| Figura 4.3-40: Pasarela del servicio web del generador de funciones .....          | 105 |
| Figura 4.3-41: Estructura del sistema de gestores .....                            | 106 |
| Figura 4.3-42: VI array instrumentos.....  | 107 |
| Figura 4.3-43: Interface Servidor.....   | 108 |
| Figura 4.3-44: Código del Gestor de Control .....                                  | 109 |
| Figura 4.3-45: Código del Gestor Generador.....                                    | 111 |
| Figura 4.3-46: Código del Gestor de Mensajes.....                                  | 112 |
| Figura 4.4-1: Flujograma del sistema software .....                                | 113 |
| Figura 5.2-1: Visor Firestorm 1 .....  | 119 |
| Figura 5.2-2: Visor Firestorm 2 .....  | 119 |
| Figura 5.2-3: Visor Firestorm 3 .....  | 120 |
| Figura 5.2-4: Visor Firestorm 4 .....  | 120 |
| Figura 5.2-5: Avatar editando apariencia .....                                     | 121 |
| Figura 5.2-6: Controles de movimiento.....   | 121 |
| Figura 5.2-7: Registro del chat.....   | 122 |
| Figura 5.2-8: Mensajes del chat público.....                                       | 122 |
| Figura 5.2-9: Mensaje por chat privado .....                                       | 123 |
| Figura 5.2-10: Controles de cámara .....   | 124 |
| Figura 5.2-11: Imagen de entrenamiento 1.....                                      | 124 |
| Figura 5.2-12: Imagen de entrenamiento 2.....                                      | 125 |
| Figura 5.2-13: Imagen de entrenamiento 3.....                                      | 125 |
| Figura 5.2-14: Imagen de entrenamiento 4.....                                      | 126 |
| Figura 5.2-15: Imagen de entrenamiento 5.....                                      | 126 |
| Figura 5.2-16: Sala de estar .....   | 127 |
| Figura 5.2-17: Botón para abrir el inventario .....                                | 127 |
| Figura 5.3-1: Vista aérea de la isla.....  | 129 |

|   |     |
|---|-----|
| Figura 5.3-2: Vista del exterior del edificio .....                           | 130 |
| Figura 5.3-3: Edificio de electrónica .....                                   | 130 |
| Figura 5.3-4: Panel de reservas .....   | 131 |
| Figura 5.3-5: Panel informativo del laboratorio .....                         | 132 |
| Figura 5.3-6: Sala de video .....   | 133 |
| Figura 5.3-7: Botón para iniciar videos .....                                 | 133 |
| Figura 5.3-8: Sala de I+D.....  | 133 |
| Figura 5.3-9: Aula de laboratorio .....                                       | 134 |
| Figura 5.3-10: Reproductor de diapositivas .....                              | 134 |
| Figura 5.3-11: Botones del reproductor.....                                   | 135 |
| Figura 5.3-12: Puesto de laboratorio .....                                    | 135 |
| Figura 5.3-13: Monitor informativo .....                                      | 136 |
| Figura 5.3-14: Ejemplo de instrumento .....                                   | 137 |
| Figura 5.3-15: Placa del divisor de tensión.....                              | 139 |
| Figura 5.3-16: Esquemático del divisor de tensión .....                       | 139 |
| Figura 5.3-17: Cajones de componentes.....                                    | 140 |
| Figura 5.3-18: Cajones de componentes abiertos .....                          | 141 |
| Figura 5.3-19: Resistores y placa divisor de tensión .....                    | 142 |
| Figura 5.3-20: Divisor de tensión con resistores.....                         | 142 |
| Figura 5.3-21: Cajón de cables .....  | 143 |
| Figura 5.3-22: Cables fuente de alimentación .....                            | 144 |
| Figura 5.3-23: Conexión de los cables de la fuente de alimentación.....       | 145 |
| Figura 5.3-24: Fuente de alimentación .....                                   | 146 |
| Figura 5.3-25: Conexión de los cables de la fuente de alimentación.....       | 147 |
| Figura 5.3-26: Habilitación de la fuente de alimentación .....                | 148 |
| Figura 5.3-27: Multímetro .....   | 149 |
| Figura 5.3-28: Tipos de medida del multímetro.....                            | 149 |
| Figura 5.3-29: Botón Shift del multímetro .....                               | 149 |
| Figura 5.3-30: Conectores del multímetro .....                                | 150 |
| Figura 5.3-31: Conexión del multímetro a la entrada de la placa .....         | 152 |
| Figura 5.3-32: Conexión del multímetro a la salida de la placa .....          | 153 |
| Figura 5.3-33: Placa con el puente 1 quitado.....                             | 154 |
| Figura 5.3-34: Medida de corriente con el multímetro .....                    | 155 |
| Figura 5.3-35: Generador de señales .....                                     | 156 |
| Figura 5.3-36: Formas de onda .....   | 156 |
| Figura 5.3-37: Parámetro del generador.....                                   | 157 |
| Figura 5.3-38: Teclado numérico del generador de señales .....                | 157 |
| Figura 5.3-39: Botones para la elección de las unidades del generador .....   | 158 |
| Figura 5.3-40: Habilitación de la salida del generador.....                   | 159 |
| Figura 5.3-41: Cable del generador de funciones .....                         | 160 |
| Figura 5.3-42: Conexión del generador como excitación de la placa .....       | 160 |
| Figura 5.3-43: Osciloscopio .....   | 161 |
| Figura 5.3-44: Pantalla del osciloscopio .....                                | 162 |
| Figura 5.3-45: Rueda para el ajuste de la base de tiempos .....               | 162 |
| Figura 5.3-46: Botones de configuración de los canales del osciloscopio ..... | 163 |

|   |     |
|---|-----|
| Figura 5.3-47: Botones de configuración del trigger .....       | 164 |
| Figura 5.3-48: Botón para el cambio de acoplamiento .....       | 165 |
| Figura 5.3-49: Botón para el menú de medidas .....              | 165 |
| Figura 5.3-50: Botones de selección del tipo de medida .....    | 166 |
| Figura 5.3-51: Tipos de medida disponibles .....                | 166 |
| Figura 5.3-52: Sondas de los canales 1 y 2 .....                | 167 |
| Figura 5.3-53: Activación del canal 1 .....                     | 167 |
| Figura 5.3-54: Conexión del canal 1 a la placa .....            | 168 |
| Figura 5.3-55: Señal a la entrada del circuito .....            | 169 |
| Figura 5.3-56: Señal de entrada con acoplamiento en AC .....    | 169 |
| Figura 5.3-57: Señal de entrada con acoplamiento en DC .....    | 170 |
| Figura 5.3-58: Medidas a la entrada y salida del circuito ..... | 171 |
| Figura 5.3-59: Formas de onda obtenidas .....                   | 171 |
| Figura 5.3-60: Medida de pico-pico a la salida .....            | 172 |
| Figura 5.3-61: Medida del valor medio a la salida .....         | 173 |
| Figura 5.3-62: Medida del valor eficaz a la salida .....        | 174 |
| Figura 6.1-1: Figura 1 Manual Servidor .....                    | 179 |
| Figura 6.1-2: Configuración del servidor laboratorio .....      | 180 |
| Figura 6.1-3: Figura 2 Manual Servidor .....                    | 182 |
| Figura 6.1-4: Ventana de la aplicación servidor iniciada .....  | 183 |
| Figura 9.1-1: Visor Phoenix 1 .....                             | 197 |
| Figura 9.1-2: Visor Phoenix 2 .....                             | 198 |
| Figura 9.1-3: Visor Phoenix 3 .....                             | 198 |
| Figura 9.1-4: Visor Phoenix 4 .....                             | 199 |
| Figura 9.2-1: Visor SecondLife 1 .....                          | 200 |
| Figura 9.2-2: Visor SecondLife 2 .....                          | 200 |



# INDICE DE TABLAS

Tabla 2.2-1: Características de los diferentes tipos de laboratorios

Tabla 4.2-1: Comparativa entre sistemas operativos

Tabla 7.1-1: Coste Software

Tabla 7.2-1: Coste Hardware

Tabla 7.3-1: Coste de ingeniería

Tabla 7.4-1: Coste total



# **1** INTRODUCCIÓN, DESCRIPCIÓN Y OBJETIVOS



## 1.1 Introducción

Las prácticas en laboratorios forman una parte muy importante de la formación en todos los programas docentes, ya que éstas permiten a los alumnos una demostración y aprendizaje de los conceptos e ideas mediante instrumentos y situaciones reales.

A pesar de esta importancia, la creación de un laboratorio no es una tarea fácil, ya que el hecho de equipar un laboratorio (según su campo de aplicación) puede suponer un gran gasto económico, tanto inicial como posterior a causa de su mantenimiento y operación.

Los estudiantes suelen encontrarse con otros problemas a la hora de realizar las prácticas en los laboratorios. Muchas veces no es posible el acceso a los laboratorios, bien por su saturación por el número de alumnos fuera del horario de clases o por encontrarse cerrados; laboratorios con falta de instrumentación; desplazarse hasta el laboratorio para realizar unas prácticas que en muchos casos se tarda menos en obtener los datos, que en lo que se tarda en realizar el trayecto y la preparación de la misma, etc....

Como solución a parte de esta problemática descrita, surge la educación a distancia, y en concreto los laboratorios virtuales, es decir, simulaciones de un laboratorio real utilizando modelos matemáticos. Por sus características y flexibilidad se han ido desarrollando laboratorios virtuales en el ámbito docente, pero no todas las áreas cuentan con tantas posibilidades o facilidades como en la electrónica.

La mayoría de los laboratorios accesibles desde Internet que hay en la actualidad dentro de la enseñanza a distancia o formación online, son virtuales. El laboratorio que se ha desarrollado tiene como principal ventaja la realización de prácticas controlando instrumentos y circuitos reales de forma remota.

Gracias al avance tecnológico desarrollado hasta la actualidad, los laboratorios remotos pueden desarrollarse como educación a distancia con suficientes garantías de éxito.

Para el desarrollo de esta aplicación software se plantea el desarrollo de una serie de tareas que se irán abordando en los diferentes capítulos del proyecto.

En el capítulo dos se abordará el estudio de los diferentes laboratorios existentes (virtuales, remotos...), sus componentes mas típicos, y en la última parte de dicho capítulo un análisis de algunos ejemplos de laboratorios remotos desarrollados por diferentes instituciones educativas.

En el capítulo tres se expone la solución escogida para realizar la aplicación software del laboratorio remoto, tras hacer un estudio de las posibilidades de implementación.

Por último, en el capítulo cuatro, se detalla la solución de la aplicación software implementada para completar el laboratorio remoto en el área de la electrónica analógica.

## 1.2 Objetivos

Aprovechando las nuevas tecnologías de la información y comunicación, el objetivo de este proyecto es desarrollar un sistema software que permita la utilización de instrumentos reales de forma remota que, junto con un hardware basado en módulos analógicos programables, cree un laboratorio remoto como complemento a la formación presencial tradicional en el área de la electrónica analógica.

Para conseguir alcanzar el objetivo principal, se van a definir una serie de objetivos secundarios que dependen de aspectos relevantes relacionados con el diseño e implementación de un laboratorio remoto. Estos objetivos son:

- La aplicación software de ser **escalable**, se debe permitir que el laboratorio sea fácilmente ampliable.
- El sistema debe ser **versátil**, cuantas más posibilidades ofrezca el sistema, mayor será su aprovechamiento.
- La aplicación debe ser fácil de **depurar**.
- Posibilitar el acceso a los recursos de un laboratorio a un mayor número de alumnos.

## 1.3 Descripción del proyecto

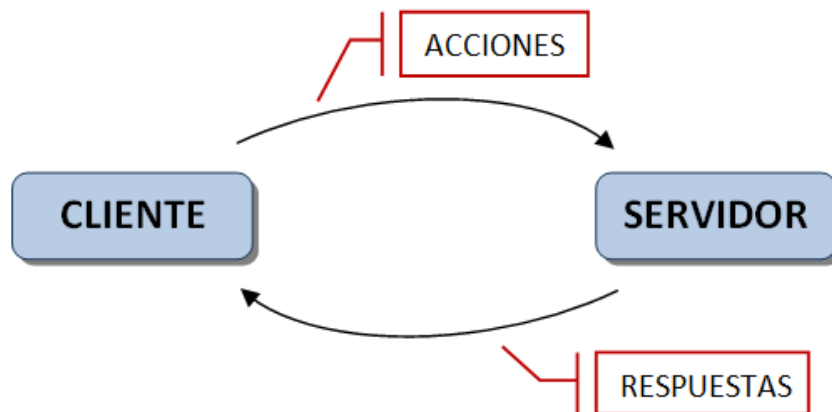
El proyecto consiste en realizar una aplicación software para implementar un laboratorio remoto en el área de la electrónica analógica, que pueda ser utilizado como complemento a las actividades formativas que se realizan en los laboratorios de los centros de enseñanza (tanto secundaria, formación profesional o universitarios) y por profesionales relacionados con el ámbito de la electrónica para realizar actividades relacionadas con su formación continua o aprendizaje permanente (*Life-long Learning*).

El sistema completo también consta de un hardware controlado mediante buses de comunicación estándar, que permite la implementación de distintos circuitos analógicos, de tal forma que se pueda realizar prácticas sobre circuitos físicos reales.

Como se ha comentado anteriormente, este laboratorio destaca por manejar de forma remota instrumentos y circuitos reales. Para ello cuenta con un sistema de comunicación basado en un modelo cliente-servidor.

Para desarrollar un laboratorio lo más real posible, la aplicación que maneja el estudiante es un **visor 3D**. Con la utilización de un visor 3D lo que se pretende es tener un aumento de la realidad a la hora de realizar las prácticas de laboratorio remotamente.

La arquitectura del sistema software que se ha implementado en el laboratorio remoto se muestra en la Figura 1.3-1:



**Figura 1.3-1: Sistema software del laboratorio remoto**

Los elementos clave que constituyen la plataforma cliente-servidor son los siguientes:

- **Servidor:** se encarga de procesar las acciones que realiza el cliente y controla y monitoriza los instrumentos y dispositivos del sistema hardware.
- **Cliente:** sería el usuario final, que mediante un visor 3D comunica las acciones a realizar al servidor para que este las procese.



# 2

## LABORATORIO REMOTO



## 2.1 Introducción

En la actualidad el desarrollo de laboratorios remotos está sufriendo una gran evolución. Su aplicación no es uniforme en todas las áreas de la ingeniería debido fundamentalmente al tipo de experimentación o prácticas que se pueden realizar en cada disciplina.

Por este motivo y como punto de partida para abordar el estudio de los laboratorios remotos puede ser interesante comenzar con un análisis de los tipos y tipologías existentes de laboratorios, así como la estructura y requerimientos de los mismos que deberían tenerse en cuenta para obtener una idea global de las posibles ventajas de su realización.

En la última parte del capítulo se encuentran una serie de ejemplos de laboratorios remotos relacionados con el ámbito de la electrónica analógica.

## 2.2 Tipos de laboratorios

Las actividades prácticas en el área de ingeniería se realizan en los laboratorios, donde el procedimiento más habitual se basa en la interacción directa con los dispositivos e instrumentos. A este tipo de laboratorios se les denomina **laboratorios tradicionales o presenciales**.

Con la creciente integración del ordenador a este tipo de laboratorios se facilita, por un lado, el control de dispositivos o instrumentos mediante la utilización de algún interface y, por otro, la simulación, mediante los modelos adecuados, del experimento completo o ciertas fases del mismo.

Junto con el uso del ordenador está la disponibilidad de uso de Internet, lo cual ha generado la aparición de nuevos tipos de laboratorios donde desarrollar las

actividades prácticas a distancia, sin la necesidad de estar presentes en un local determinado. La clasificación de estos nuevos tipos de laboratorios puede ser la siguiente:

**1) Laboratorios virtuales:** son aquellos en los que se simula un laboratorio real utilizando modelos o cálculos matemáticos que proporcionan un entorno simulado para la realización de las prácticas, es decir, no se interactúa con dispositivos o instrumentos reales. La principal ventaja es la posibilidad de desarrollar los experimentos bajo diferentes condiciones y de forma repetitiva, sin que exista riesgo para el equipamiento utilizado. Por otro lado, los laboratorios virtuales están limitados por el modelo y para ser manejables estos tienden a simplificarse, y de esa forma se pierde información con respecto al sistema real.

**2) Laboratorios remotos:** es un laboratorio real controlado a distancia mediante alguna aplicación o a través de la Web, interaccionado directamente con los dispositivos e instrumentos allí alojados. En estos laboratorios será necesario monitorizar las acciones durante el desarrollo de la práctica para evitar cualquier posible daño en los dispositivos o instrumentos.

Contrario al laboratorio virtual los laboratorios remotos permiten la interacción directa entre el usuario y el equipo de laboratorio en tiempo real, lo que aporta una experiencia significativa en el proceso de enseñanza-aprendizaje del usuario, en el desarrollo de habilidades y competencias e incluso como valor agregado en el proceso de formación.

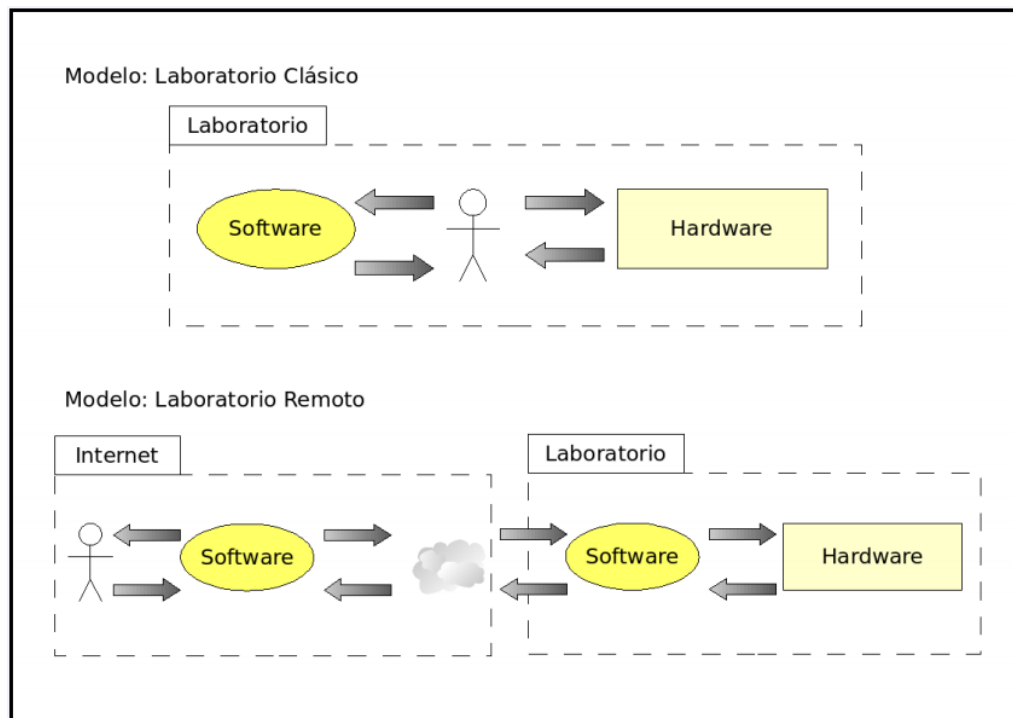
**3) Laboratorios híbridos:** como su nombre indica este laboratorio combina las prestaciones de los laboratorios virtuales y remotos. Esta opción consiste en ofrecer la posibilidad de realizar ciertas partes del experimento de forma virtual o simulada en primer lugar, consiguiendo una experiencia que le ayudará a aprovechar mejor el resto del experimento haciendo uso de los dispositivos e instrumentos reales.

En la Tabla 2.2-1 se resumen las características básicas en cuanto a la ubicación del usuario y el uso de dispositivos de los diferentes tipos de laboratorios.

| Tipos de Laboratorio | Ubicación del Usuario | Uso de dispositivos |
|----------------------|-----------------------|---------------------|
| <b>Tradicionales</b> | Presencial            | Reales/Simulados    |
| <b>Virtuales</b>     | Remota                | Simulados           |
| <b>Remotos</b>       | Remota                | Reales              |
| <b>Híbridos</b>      | Remota                | Reales/Simulados    |

**Tabla 2.2-1: Características de los diferentes tipos de laboratorios**

En la Figura 2.2-1, se puede observar la estructura general de un laboratorio clásico (parte superior de la imagen) y de un laboratorio remoto (parte inferior):



**Figura 2.2-1: Comparativa entre L.Tradicional y L.Remoto**

A continuación se presenta una tabla comparativa/calificativa de costes entre el laboratorio presencial y remoto.

|                            | Laboratorio Remoto   | Laboratorio Presencial |
|----------------------------|----------------------|------------------------|
| Material de práctica       | Sí                   | Sí                     |
| PC                         | Sí                   | Sí                     |
| Software                   | Sí                   | No                     |
| Conexión a Internet        | Sí                   | No                     |
| Webcam                     | Sí                   | No                     |
| Desarrollo                 | Sí                   | No                     |
| Coste Absoluto             | Remoto > Presencial  |                        |
| Coste relativo por usuario | Remoto << Presencial |                        |

**Tabla 2.2-2: Comparativa de costes entre laboratorio Remoto y Presencial**

Algunos de los datos pueden variar según el tipo de laboratorio, pero en general los laboratorios remotos tienen los costes absolutos más altos en la mayoría de aspectos, en material, en software y hardware, en desarrollo, etc. Pero el coste relativo por usuario sería más bajo, por lo tanto valdría la pena implementar este tipo de laboratorios en la medida que sea posible.

La creciente complejidad de las actividades prácticas en laboratorios y el desarrollo de las TIC, han hecho que los Laboratorios Virtuales evolucionen, transformándose en laboratorios remotos, ya que éstos presentan mayores ventajas, debido a que proporcionan un mayor nivel de interactividad y el alumno entra en contacto con equipamiento real, en lugar de programas simulados.

A la hora de definir las ventajas y limitaciones, el criterio a comparar son los diferentes aspectos entre los laboratorios tradicionales y los laboratorios remotos, con este tipo de laboratorio se tienen características sobresalientes como:

- El estudiante o usuario trabaja con instrumentos reales, no con programas de simulación.
- Acceso desde cualquier ubicación que disponga Internet con un mayor rendimiento de los equipos del laboratorio, por su disponibilidad 24 horas al día 7 días a la semana.

- Organización flexible de los laboratorios. No es un espacio que deba estar abierto a todas horas, y que deba ser controlado por los profesores, basta con que el acceso vía Internet esté operativo.
- Organización flexible del trabajo de los alumnos. Con los laboratorios remotos lo alumnos y profesores pueden organizar mejor su tiempo, al igual que los horarios de clase.
- Aprendizaje autónomo. Los laboratorios remotos fomentan el trabajo autónomo, algo fundamental en el nuevo espacio de educación superior.
- Otras modalidades de formación. Los laboratorios remotos permiten organizar cursos no presenciales o semi-presenciales, pensando en posibles futuros estudiantes.
- Potencial disminución de costos asociados a mantenimiento y reposición de equipos de laboratorio. Los equipos de laboratorio son protegidos de posibles averías físicas, aumentando su vida útil y disminuyendo la inversión en reemplazo de equipos.
- Posibilidad de abrir cursos en diferentes sedes, sin incurrir en gastos de desplazamiento de equipos, amplía la posibilidad de tener más cursos abiertos en el centro.
- Posible disminución en costes asociados a salas de laboratorio, espacios de almacenamiento y personal administrativo de equipos de laboratorio, que puede impactar positivamente en los costes y márgenes de los cursos.
- Integración de discapacitados. Toda vez que los elementos hardware pasan a ser controlados por un ordenador pueden ser gobernados utilizando técnicas software/hardware para discapacitados.

Los aspectos negativos existentes se pueden resumir en las siguientes:

- Falta de apoyo, no hay un profesor que aclare dudas o ejemplifique procedimientos.
- Se necesita cierta destreza en el manejo de las tecnologías.
- Dependiendo de las posibilidades de la aplicación de acceso al laboratorio remoto puede existir ausencia de presencia física del profesor y compañeros de estudios.

- Dependencia de la fiabilidad y rapidez de la conexión a Internet.
- La eficacia de los laboratorios remotos dependerá del área en el que se desarrollen. No será la misma para ingeniería informática que para ingeniería química.
- La interacción del alumno con el experimento e instrumentos que se utilizan en las diferentes prácticas está limitada.

## 2.3 Tipologías de laboratorios remotos

Los laboratorios remotos todavía no están lo suficientemente extendidos, por lo que no hay una estructura o tipología asentada.

Antes de describir la estructura y componentes de un laboratorio remoto aquí se presentan algunos criterios para clasificar los diferentes tipos de laboratorios remotos.

1.- En función del acceso al experimento o práctica con el que se va a trabajar:

- **Acceso exclusivo:** Los estudiantes tienen el control de los dispositivos de forma exclusiva durante un intervalo de tiempo. Este caso se suele dar cuando se utilizan recursos con un elevado coste o en los que los procesos llevados a cabo en el desarrollo de la práctica requieren una respuesta temporal lenta.

- **Acceso simultáneo:** Los estudiantes no tienen restricciones temporales de acceso para comunicarse con los dispositivos. Varios estudiantes (dentro de un límite) pueden interaccionar con el puesto remoto en el mismo instante, sin que perciban que están compartiendo los recursos. Se asocia este tipo de laboratorio a experimentos en los que los costes permiten la posibilidad de utilizar recursos redundantes o en los que las respuestas temporales de las prácticas sean rápidas.

2.- En función de la interacción con los experimentos o prácticas a desarrollar:



- **Indirecta** (batch type): Las acciones para controlar o desarrollar las prácticas son ejecutadas en el laboratorio remoto sin la interacción directa del estudiante y los resultados obtenidos son remitidos con posterioridad al mismo.
- **Directa** (interactive type): Los usuarios tienen la posibilidad de comunicarse directamente con los dispositivos e instrumentos mientras están desarrollando la práctica obteniendo los resultados en tiempo real.
- **De monitorización** (sensor type): Los usuarios sólo reciben los resultados procedentes de sensores o equipos para su monitorización o procesamiento. No hay interacción con el experimento o práctica.

## 2.4 Análisis de requisitos para la implementación

En cuanto a los requerimientos existentes para la implementación, se pueden diferenciar dos niveles: requisitos de comunicación y acceso, y requisitos por parte de la gestión del laboratorio.

En cuanto a los relacionados con la comunicación y el acceso:

- Alto nivel de seguridad para evitar posibles interacciones que dañen los dispositivos hardware que se utilicen.
- Control del estado del experimento en cada instante que asegure, en un acceso multiusuario, la integridad del desarrollo de la práctica que esté realizando cada usuario.
- Los protocolos y puertos a utilizar en las comunicaciones entre el usuario y el laboratorio remoto deben asegurar que no existan problemas con los firewalls o antivirus instalados en los ordenadores.
- La interacción del usuario debe realizarse sin retardos o interrupciones que impidan la correcta consecución de los objetivos de aprendizaje planteados.

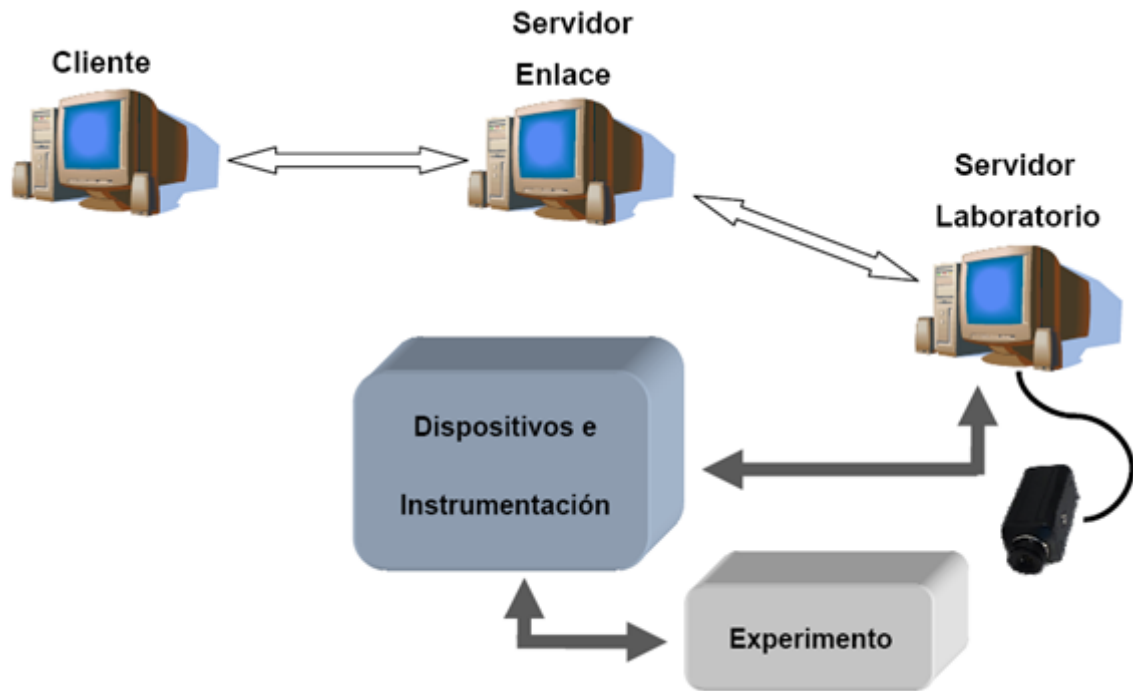
Dentro de los referentes por parte de la gestión del laboratorio estarían incluidos:

- Permitir un registro de usuarios para tener control sobre laboratorios
- Tener un sistema de reserva de los laboratorios
- Utilizar herramientas de código abierto
- Tener un sistema de ayuda donde se tengan manuales, guías, tutoriales disponibles para los usuarios del sistema
- Construir el sistema con diferentes niveles de acceso dependiendo el tipo y el perfil de los usuarios

Tras observar estos requerimientos básicos se puede extraer como conclusión que la implementación de un laboratorio remoto es un asunto complejo en el que las capacidades técnicas y formativas deben ser convenientemente planificadas, coordinadas y utilizadas.

## 2.5 Estructura y componentes de un laboratorio remoto

Una vez explicada la tipología existente, la estructura básica que aparece en la mayoría de los laboratorios remotos está formada por los siguientes elementos (Figura 2.5-1):



**Figura 2.5-1: Componentes típicos de un laboratorio remoto**

- **Ordenador cliente:** el estudiante para poder interaccionar con el experimento necesita utilizar algún tipo de recurso que se ejecute en su ordenador. Estos recursos podrían ser aplicaciones de programación C, C++, Java, Visual Basic, .NET, Python, LabView, una aplicación Web, visor 3D, etc.

- **Dispositivos e instrumentos:** para poder interaccionar con el experimento se necesita un hardware específico, el cual permitirá básicamente el control de dicho experimento y la obtención de resultados. En cuanto a los interfaces que se utilizan para permitir la comunicación entre la aplicación que se ejecuta en el servidor del laboratorio, con los dispositivos e instrumentos es mayoritario el uso de buses estándar (IEEE-488, USB, RS-232, PXI, LXI, etc.) aunque también es posible hacer uso de interfaces desarrollados a medida basados generalmente en tarjetas de adquisición de datos.

- **El experimento:** con el que se va a trabajar en el laboratorio.

- **Servidor laboratorio:** controla y monitoriza todo el desarrollo del experimento.

También existen varias alternativas para implementar la aplicación del servidor del laboratorio como paquetes software profesionales (LabView, VEE, MATLAB/Simulink...) o una aplicación basada en el uso de lenguajes de programación de propósito general (C, C++, Python, etc).

- **Servidor de enlace:** la función de éste es interconectar al estudiante con el servidor del laboratorio. No sería necesario si existe una comunicación directa entre el ordenador del cliente y el servidor del laboratorio. En este servidor se ejecutan los recursos necesarios para interaccionar con el laboratorio remoto, y también complementarios, como autenticación de usuarios, gestión de reservas, etc.

## 2.6 Ejemplos de laboratorios remotos

A continuación se describirán algunos de los laboratorios remotos más representativos, que han ayudado a encontrar una solución hardware- software que posteriormente se detallará. Las características que se analizan se centran en la tipología de acceso (directa, indirecta o monitorización), como es el acceso del usuario y la tecnología software utilizada en su implementación.

### WebLab-Deusto

**URL:** <https://www.weblab.deusto.es/web/>

La universidad de Deusto creó una arquitectura y conjunto de laboratorios remotos con el nombre de WebLab-Deusto. Esta arquitectura fue de las primeras en aplicar Web 2.0 y en adoptar la Arquitectura Orientada a Servicios, a través de su adaptación a los laboratorios remotos conocida como SOLA. Dispone de un laboratorio remoto genérico preparado para integrar nuevos experimentos en él con un esfuerzo pequeño.



**Figura 2.6-1: Web Laboratorio WebLab Deusto**

WebLab-Deusto ha sido desplegado para distintos tipos de experimentos:

- WebLab-DEUSTO-FPGA: Permite la programación y control del dispositivo programable FPGA.
- WebLab-DEUSTO-CPLD: Permite la programación y control del dispositivo programable CPLD.
- WebLab-DEUSTO-PIC: Permite la programación y control del dispositivo programable microcontrolador PIC.
- WebLab-DEUSTO-VISIR: Permite la construcción y análisis de circuitos analógicos básicos.
- WebLab-DEUSTO-LXI: Permite la construcción y análisis de circuitos analógicos bajo el estándar LXI.
- SecondLab: Permite el control y programación de microrrobot a través del entorno SecondLife, de esta forma se aúna dos ámbitos relevantes en la educación actual.

Algunas de las principales características de este laboratorio son:

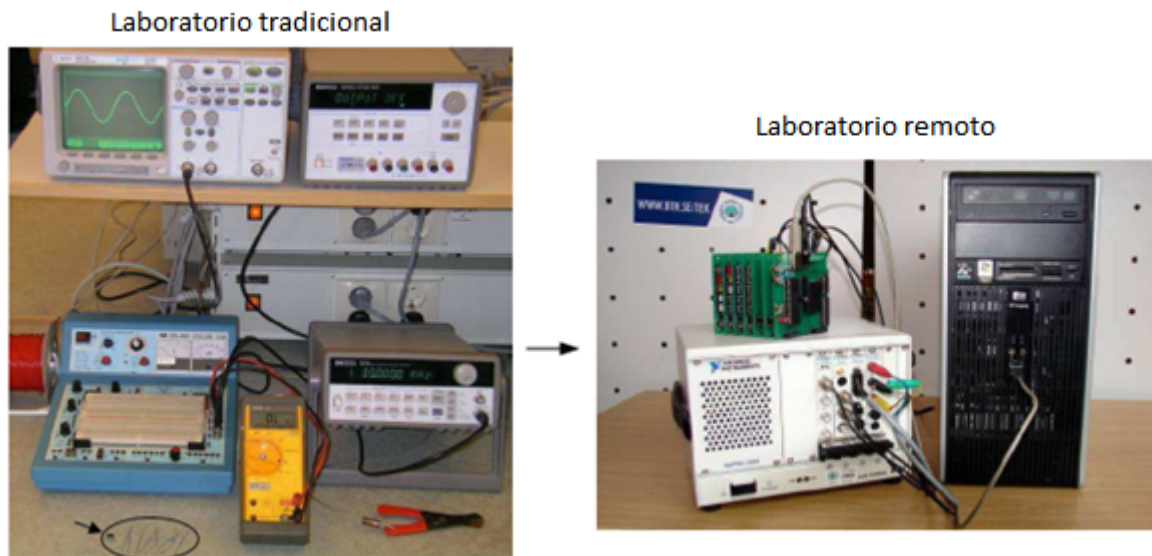
- Tipologías de laboratorio remoto: de interacción directa y de acceso exclusivo.
- Acceso del usuario: através de un navegador Web. Se requiere reserva para hacer uso del laboratorio remoto.
- Tecnologías y herramientas software utilizadas en su implementación: para el lado del cliente (JavaScript, Flash, applets Java), y para el lado del servidor (WebLab-Deusto incluye bibliotecas para C / C + +, NET, LabVIEW, Java y Python).

### Proyecto VISIR (Virtual Instrument Systems in Reality)

**URL:** <https://weblab-visir.deusto.es/electronics/>

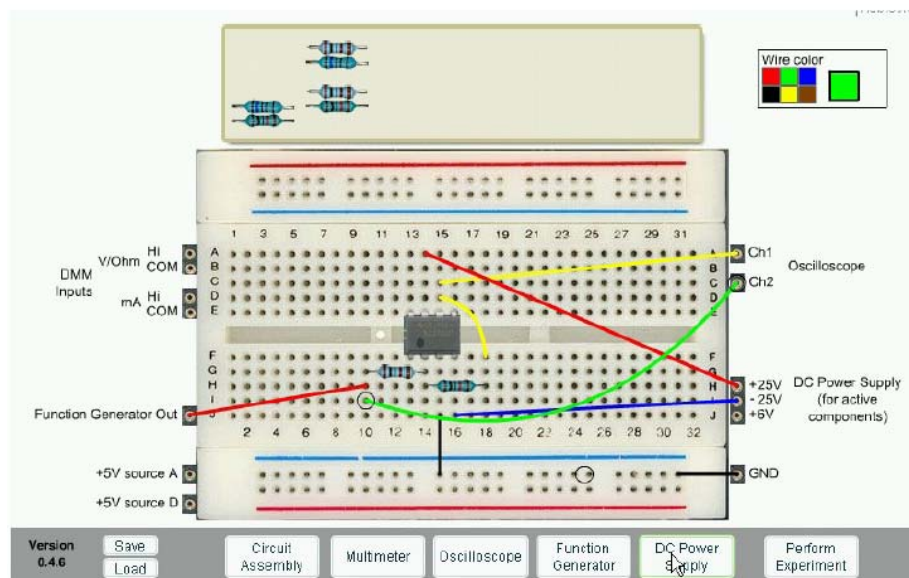
El proyecto VISIR fue iniciado, en 2006, por el Instituto de Tecnología Blekinge de Suecia y la empresa National Instrument. El objetivo de este proyecto era crear laboratorios distribuidos en línea en colaboración con otras universidades (Deusto, etc.). Para ello se pretendía desarrollar una “mesa de trabajo o workbench” on line que permitiera al estudiante realizar experimentos eléctricos desde su casa.

Dos de los aspectos más destacables de este laboratorio son: proporciona una interfaz gráfica con un gran número de dispositivos electrónicos (muy detallada y fácil de completar, añadir elementos) y la posibilidad de que varios estudiantes puedan realizar sus prácticas de forma simultánea (conurrencia).



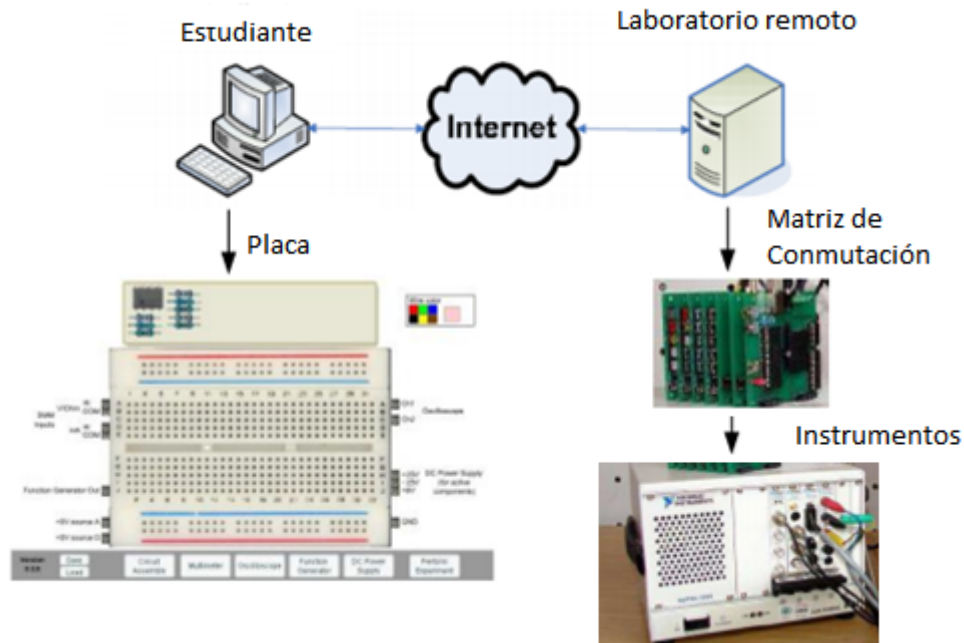
**Figura 2.6-2: Mesa de trabajo tradicional y on-line**

Como se puede ver en la Figura 2.6-2, la mayoría de los dispositivos electrónicos pueden ser controlados remotamente, pero ¿qué pasa con la placa de conexiones y los cables que se colocarán sobre ella? Para imitar esto, el instituto tecnológico de Blekinge ha diseñado una matriz de conmutación con relés electromecánicos que simulan dicha placa de conexiones y cables. En la Figura 2.6-3 se muestra como es dicha placa:



**Figura 2.6-3: Placa virtual del laboratorio remoto VISIR**

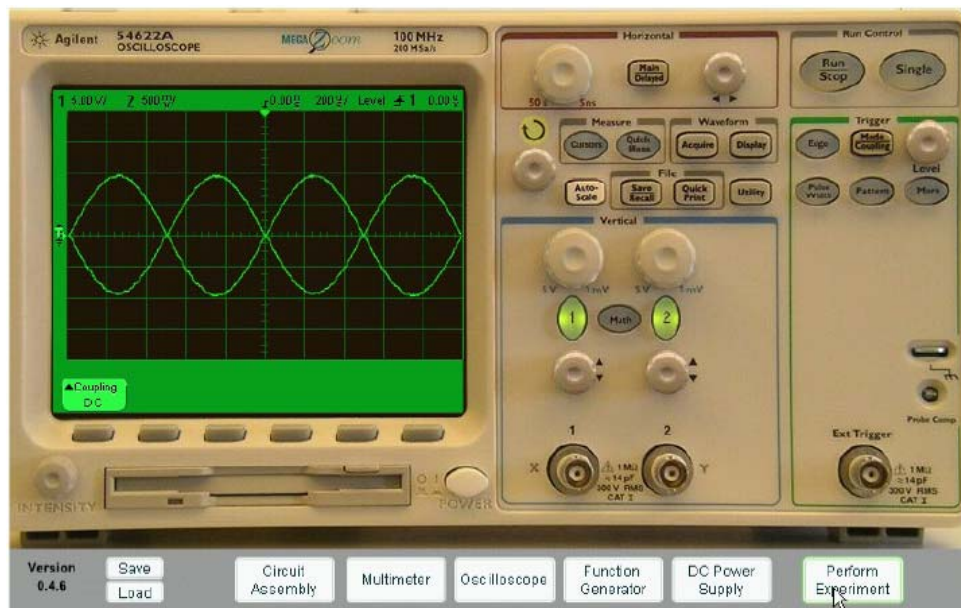
Así la estructura global sería la mostrada en la Figura 2.6-4 y consta de un PC desde donde el estudiante ve la placa simulada, el panel frontal de un osciloscopio, etc. Y un PC encargado de construir el circuito, usando la matriz de conmutación.



**Figura 2.6-4: Implementación del laboratorio remoto VISIR**

Sobre la placa el alumno puede conectar los componentes del circuito (resistores, condensadores, diodos, transistores y amplificadores operacionales) mediante cables y puede analizar el funcionamiento de dicho circuito utilizando como instrumentos una fuente de alimentación, un generador de funciones, un multímetro digital y un osciloscopio. Para manejar cada uno de los instrumentos se utiliza una representación del panel frontal de un instrumento típico de un laboratorio tradicional (Figura 2.6-5).





**Figura 2.6-5: Osciloscopio del laboratorio remoto VISIR**

La selección de diferentes circuitos a analizar y la elección de diferentes puntos de prueba en los mismos es posible gracias a una matriz de conmutación desarrollada ad hoc. Los instrumentos de excitación y medida que se pueden controlar de forma remota están insertados en un sistema PXI (Figura 2.6-6).



**Figura 2.6-6: Sistema hardware empleado en laboratorio remoto VISIR**

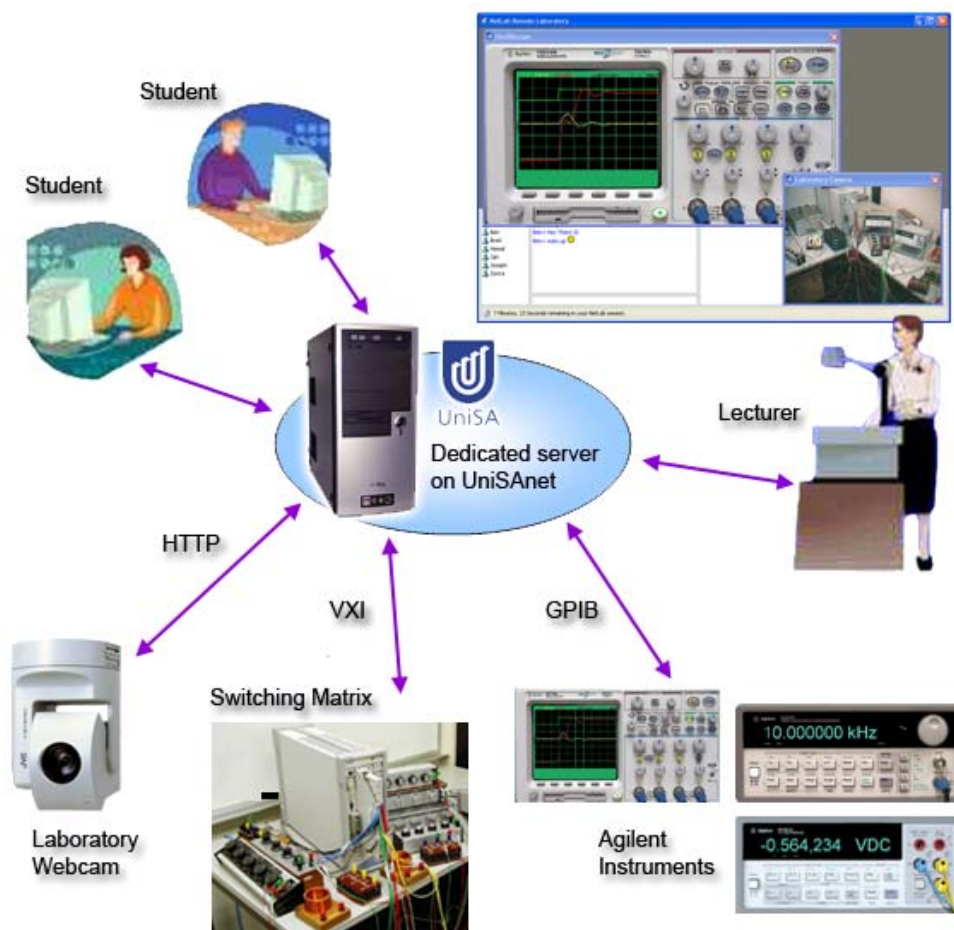
Algunas de las principales características de este laboratorio son:

- Tipologías de laboratorio remoto: de interacción directa y de acceso exclusivo.
- Acceso estudiante: a través de un navegador Web con módulo FLASH instalado. Se requiere reserva para hacer uso del laboratorio remoto.
- Tecnologías y herramientas software utilizadas en su implementación: FLASH, C++, LabView, XML, PHP y MySQL.

### NetLab (Universidad de South Australia UniSA)

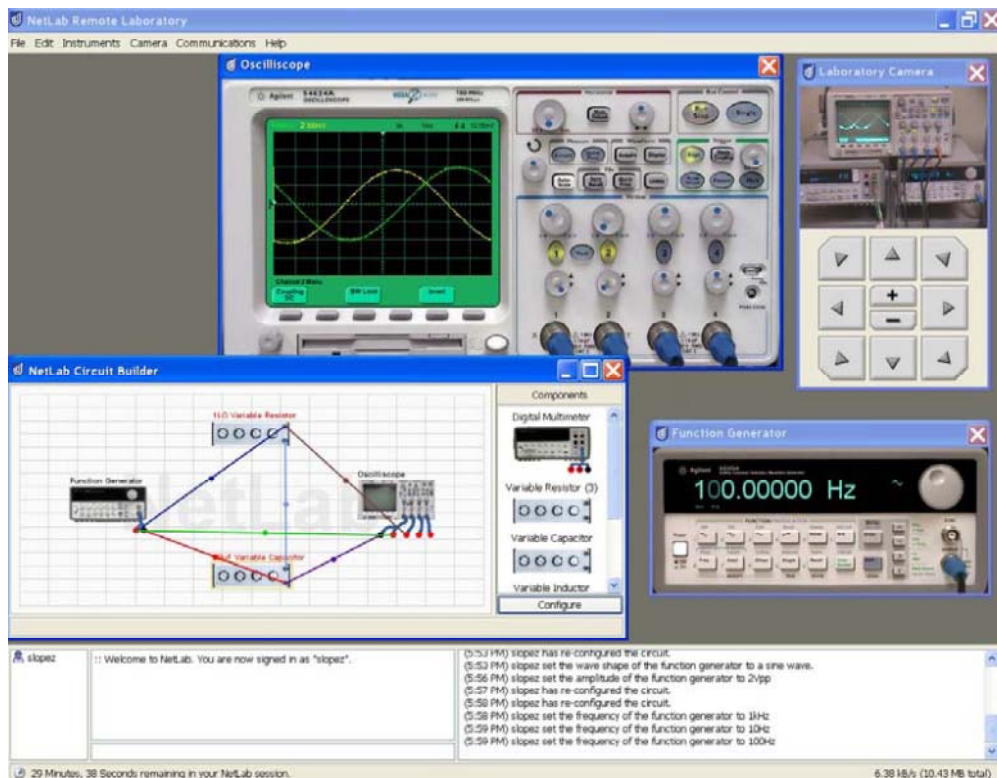
URL: <http://netlab.unisa.edu.au/index.shtml>

La estructura global del laboratorio NetLab es la mostrada en la Figura 2.6-7:



**Figura 2.6-7: Estructura global NetLab**

Permite a los usuarios realizar análisis de circuitos básicos basados en resistores, condensadores, bobinas y transformadores. El usuario diseña el circuito mediante la ayuda de un programa de edición (Circuit Builder) en el que selecciona los componentes e instrumentos a utilizar y los interconecta mediante cables. Los instrumentos disponibles son un generador de funciones, un osciloscopio y un multímetro, sobre los cuales el usuario tiene un control completo mediante un panel que muestra el panel frontal del instrumento real (Figura 2.6-8).

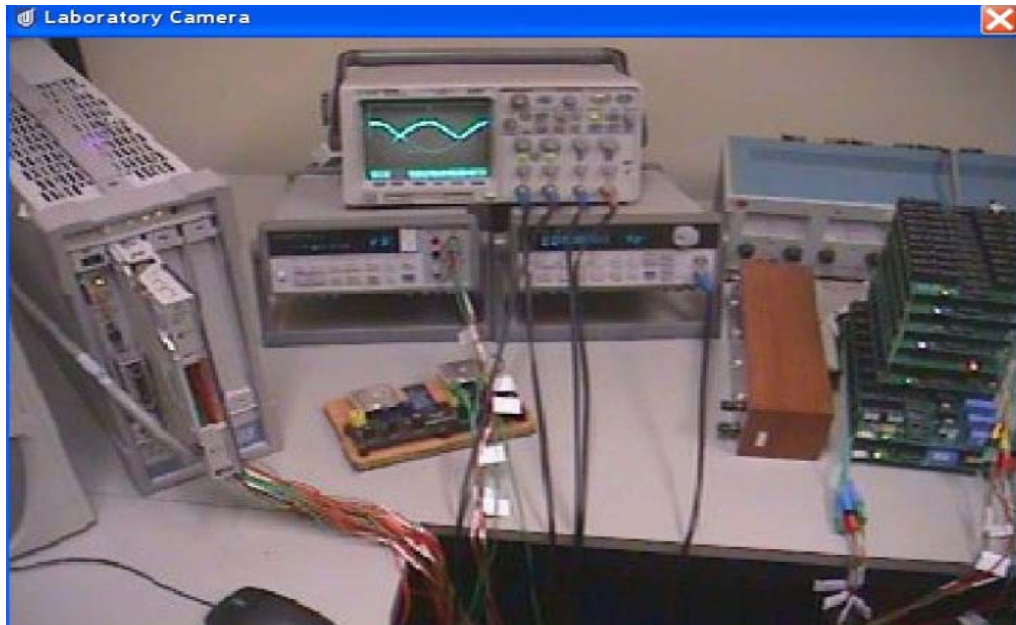


**Figura 2.6-8: Ejemplo laboratorio remoto NetLab**

Adicionalmente es interesante destacar que existe la posibilidad de realizar prácticas de forma colaborativa con hasta un máximo de tres usuarios. Todas las acciones de cada usuario son comunicadas al resto de usuarios y, por tanto, pueden ir realizando de forma conjunta todo el proceso de análisis del circuito diseñado, incluso con la posibilidad de comunicarse mediante un Chat entre ellos.

Los instrumentos utilizados se controlan mediante interface IEEE-488 mientras que una matriz de conmutación VXI y unos módulos hardware específicos permiten

la interconexión y programación de valores de los componentes pasivos utilizados (Figura 2.6-9).



**Figura 2.6-9: Equipamiento real laboratorio remoto NetLab**

Algunas de las principales características de este laboratorio son:

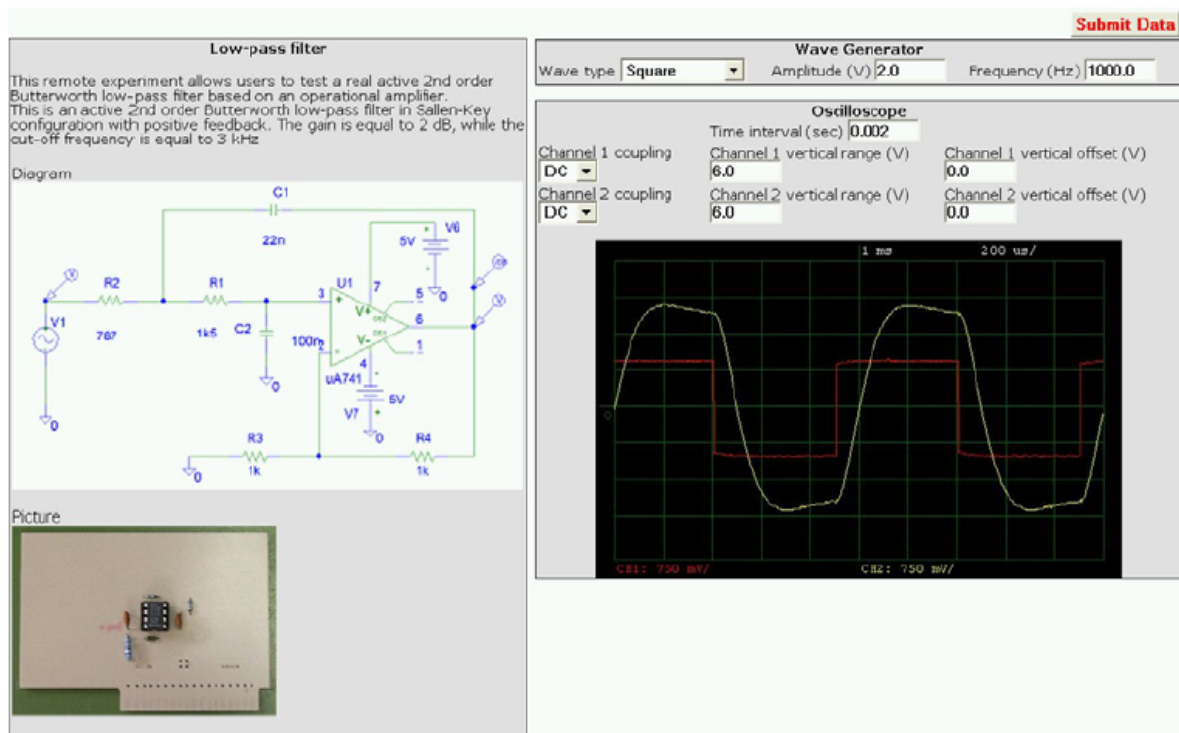
- Tipologías de laboratorio remoto: de interacción directa y de acceso simultáneo (hasta tres usuarios).
- Acceso estudiante: a través de una aplicación específica disponible para su descarga en la página Web del laboratorio remoto. Para la correcta ejecución de la aplicación es necesario tener instalado Java Runtime Environment.
- Tecnologías y herramientas software utilizadas en su implementación: en la versión más moderna: Java y VISA. En versiones anteriores: Java y LabView.



## ISILab (Internet shared Instrumentation Laboratory, Universidad de Génova, Italia)

URL: <http://isilab.dibe.unige.it/>

Permite a los usuarios analizar el funcionamiento de una serie de circuitos electrónicos analógicos y digitales disponibles en una lista de experimentos. Una vez elegido el circuito que se quiere analizar es posible comprobar el funcionamiento del mismo, programando la excitación que se estime conveniente mediante el generador de funciones y observando las señales obtenidas en los puntos de prueba especificados en el circuito con el osciloscopio (Figura 2.6-10).

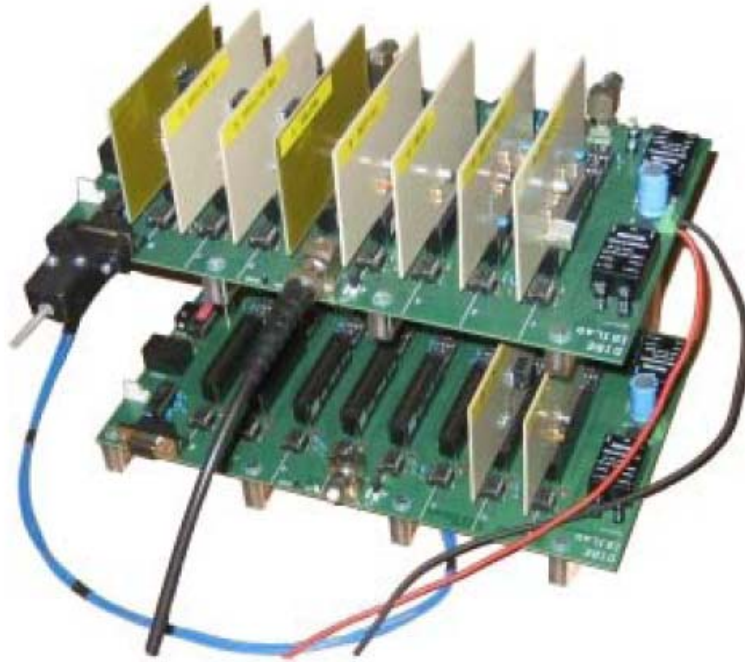


**Figura 2.6-10: Ejemplo de utilización del laboratorio remoto ISILab**

Los instrumentos utilizados pueden ser de sobremesa, controlados mediante interface IEEE-488, o implementados con una tarjeta de adquisición de datos.

Los circuitos disponibles para su análisis están integrados en una plataforma hardware que permite compartir los diferentes instrumentos de excitación y medida y elegir los puntos de salida de los circuitos mediante matrices de conmutación, de

esta forma los usuarios pueden elegir el circuito que deseen para realizar su análisis e interactuar con él al mismo tiempo que otros usuarios (Figura 2.6-11).



**Figura 2.6-11: hardware donde se integran los circuitos en el ISILab**

Algunas de las principales características analizadas de este laboratorio son:

- Tipologías de laboratorio remoto: De interacción directa y de acceso simultáneo.
- Acceso estudiante: A través de un navegador Web con Java Virtual Machine instalado.
- Tecnologías y herramientas software utilizadas en su implementación: En la versión más moderna: WSRF.NET (Web Service Resource Framework). En versiones anteriores: Java, XML y LabView.

Más ejemplos notables de laboratorios remotos:

- Automatic Control Telelab <http://act.dii.unisi.it/home.php>, con experimentos remotos sobre diferentes plantas (motor, levitador magnético, helicóptero, robot LEGO, tanques), y que permite introducir código de control desarrollado por el

usuario e incluso, contempla el desarrollo de prácticas de identificación de sistemas.

- Laboratorio Remoto de Automática industrial <http://ira.unileon.es/es>. Es un laboratorio remoto con acceso a equipos industriales, como una planta piloto para la realización de experiencias de control de operación y supervisión remota, maquetas de procesos de control sobre variables de nivel, caudal, temperatura y otros equipos de automatización.
- Mechatronics Remote-Access Laboratory: tiene diseño de controladores PID para un sistema de tercer orden construido con circuitos RC. Permite introducir el código de control diseñado por el usuario, basado en LabView. <http://mechanical.poly.edu/research/control/RemoteLab.htm>
- Web shaker: es un experimento sobre una mesa agitadora que simula terremotos, que puede ser controlado a través de Internet con el objetivo de diseñar el movimiento y evaluar sus efectos. <http://webshaker.ucsd.edu>.
- El instituto Tecnológico de Monterrey ha iniciado el proyecto de una Red Intercampus, que busca impulsar el desarrollo de plataformas de laboratorios remotos en las áreas de ingeniería eléctrica, electrónica y automatización y control. <http://www.itesm.edu/wps/wcm/connect/snc/portal+informativo/por+campus/laguna/academia/noticia+laboratorios+remotos+ing+lag>





# **3** SOLUCIÓN ESCOGIDA PARA EL LABORATORIO REMOTO



## 3.1 Introducción

Después de hacer un estudio sobre los laboratorios remotos es la hora de definir cómo se ha implementado el laboratorio remoto en el área de la electrónica analógica.

En un principio se pretende que dicho laboratorio remoto pueda ser utilizado como complemento en las enseñanzas que se imparten en la Escuela Universitaria de Ingeniería Técnica de Telecomunicación de la Universidad Politécnica de Madrid, no obstante, en un futuro cercano, es factible que su utilización se extienda a cualquier área o incluso se pueda abrir su utilización a cualquier usuario ajeno a la UPM que disponga de acceso a Internet.

La propuesta del sistema software que se va a presentar en los próximos apartados tiene como base una serie de especificaciones que se detallarán en el apartado 3.2 *Especificaciones generales*.

Una vez conocidas las especificaciones se analizarán una serie de alternativas que se han tenido en cuenta para desarrollar la arquitectura del sistema software. El objetivo de esta parte es definir un sistema lo suficientemente detallado, partiendo de la especificación y sabiendo cuales son los recursos disponibles para hacerlo.

En cuanto al sistema hardware del que se dispone para la implementación del laboratorio remoto, consta de una serie de dispositivos y un bloque de experimento.

Antes de abordar la solución escogida referente al sistema software, hay que mencionar que el bloque de experimento que se describe en el apartado 3.3 *Sistema hardware*, ya ha sido desarrollado. Este bloque consiste en un sistema hardware necesario para la configuración de las prácticas. Toda la documentación referente a él se encuentra en el Proyecto Fin de Carrera, “**Desarrollo de un sistema hardware programable para configurar circuitos analógicos básicos**”.

## 3.2 Especificaciones generales

Tanto el diseño software cómo hardware se han basado en las siguientes especificaciones:

- **Escalabilidad.** El hardware y el software deben permitir que las posibilidades del laboratorio sean fácilmente ampliables. En cuanto al hardware, esto significa que no sea necesario realizar modificaciones si se incluyen nuevos instrumentos de medida y excitación o se incrementa el número de prácticas. Respecto al software, que se puedan añadir nuevos módulos sin demasiado esfuerzo de programación. En definitiva, tanto el software como el hardware deben tener una estructura modular.
- **Versatilidad.** El hardware y el software deben ofrecer múltiples funcionalidades. En el hardware implica que el usuario tenga opciones para cambiar parámetros de los circuitos, así como para realizar distintas medidas en varios puntos de los circuitos. En cuanto al software, este debe poder controlar diferentes tipos de dispositivos sin necesidad de grandes cambios.
- **Disponibilidad y acceso simultáneo.** El laboratorio debe ser accesible a cualquier usuario a través de Internet, a cualquier hora del día y permitir que varios usuarios puedan operar al mismo tiempo.
- **Facilidad de depuración.** Tanto en la fase de implementación como en la fase de funcionamiento es necesario detectar posibles errores o fallos, e incluso contar con un módulo que sirva de depuración.
- **Minimización de la virtualidad.** Los usuarios deben tener sensación de realidad al interaccionar con los instrumentos del laboratorio. Por tanto es necesario diseñar el sistema para que el usuario sienta un control real sobre el laboratorio.
- **Coste reducido.** Para minimizar el coste se intentará aprovechar las infraestructuras existentes en los laboratorios de la escuela.

### 3.3 Sistema hardware

El sistema hardware del que se dispone posee la estructura reflejada en la Figura 3.3-1:

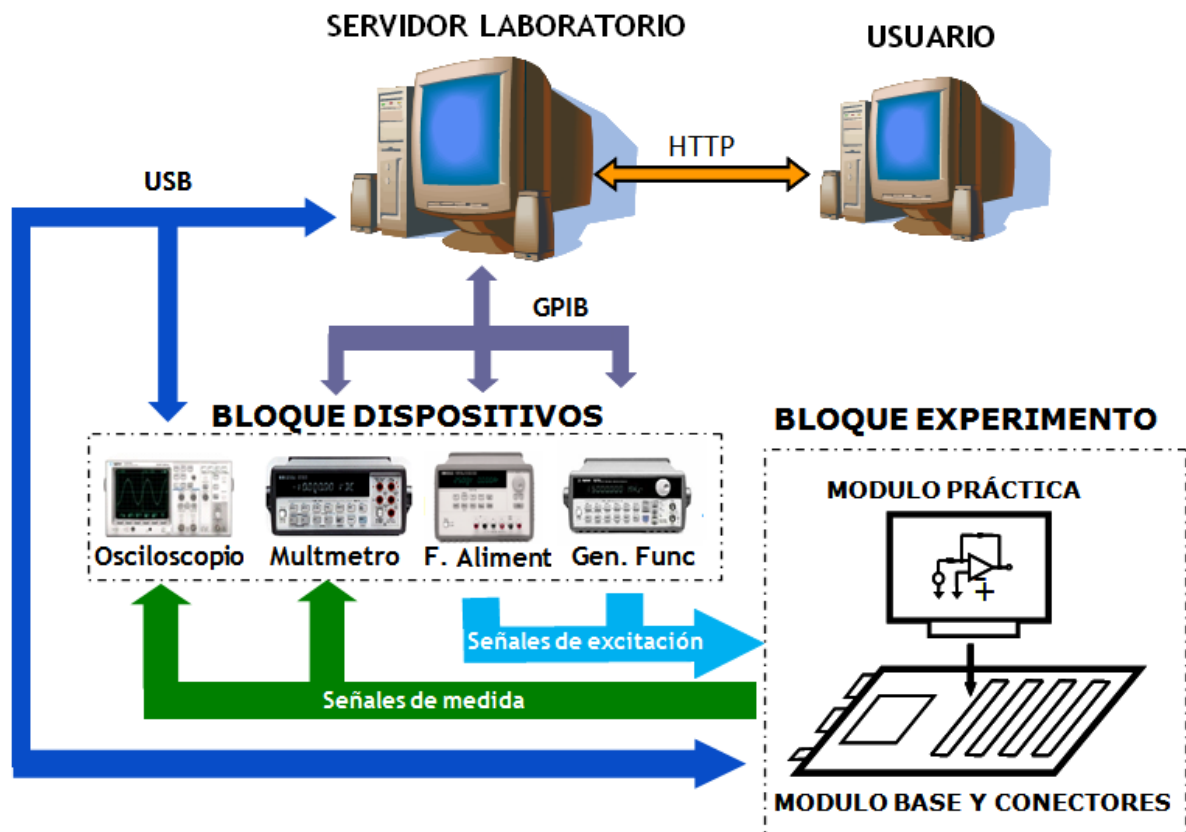


Figura 3.3-1: Sistema hardware del laboratorio remoto

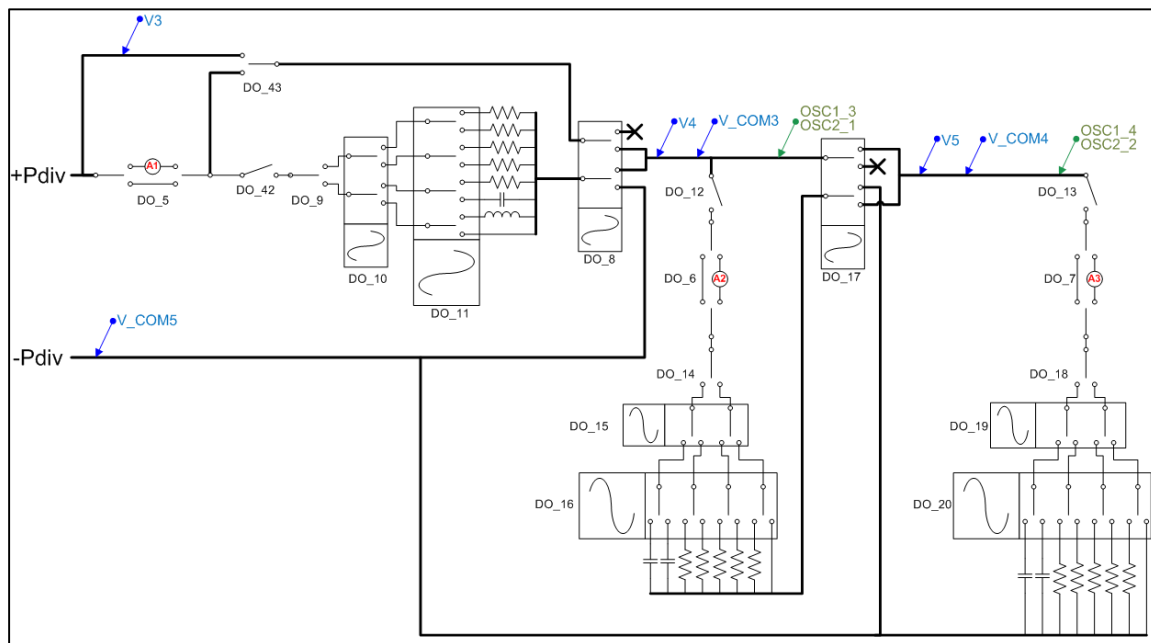
#### Bloque de Experimento

El bloque experimento incluye los circuitos analógicos que se utilizan en cada una de las prácticas que se propongan. Está constituido por un sistema modular basado en un módulo base, que se comunica mediante USB al servidor y que dispone de una serie de “slots” (módulo conectores), donde se intercambian los diferentes módulos de prácticas.

A continuación se describen brevemente las diferentes partes del bloque experimento:

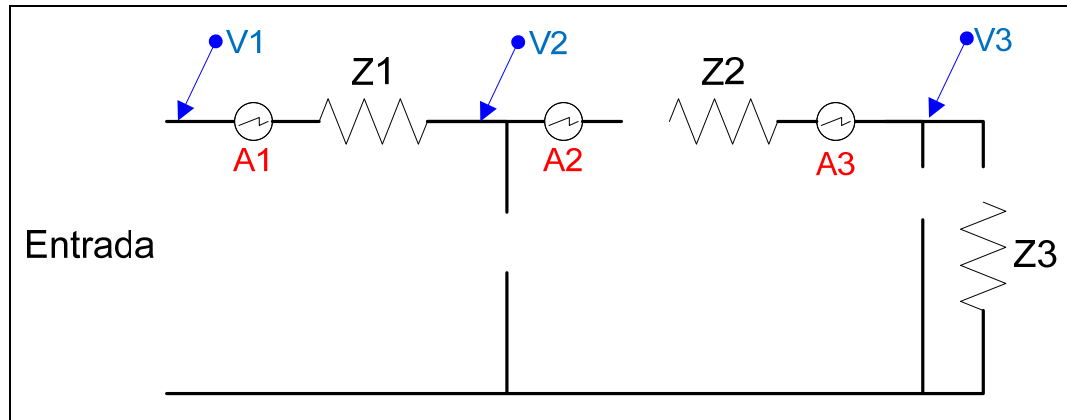
- **Módulo Base:** su función básica es la de trasladar a nivel hardware las órdenes de actuación sobre cada uno de los módulos de prácticas, recibidas desde el servidor al que está conectado a través del bus USB. Está diseñado para el control y gobierno de hasta 8 Módulos de Prácticas a la vez.
- **Módulo Conectores:** está diseñado para que se inserten en él los módulos de prácticas. Se podría asemejar a una especie de “rack”, donde en cada ranura se conectaría un módulo.
- **Módulo de Práctica:** son módulos que contienen los elementos y componentes necesarios para la realización de las distintas prácticas.

En concreto, el módulo utilizado como práctica para el desarrollo del *MANUAL DE USUARIO I: VISOR 3D* en el apartado 5, es el de módulo de *Análisis de Circuitos*. Este módulo sirve para probar el funcionamiento de las numerosas configuraciones que se pueden realizar con una serie resistencias, condensadores y bobinas, en sus distintas asociaciones. El esquema de este módulo y los componentes que presenta es el mostrado en la Figura 3.3-2:



**Figura 3.3-2: Esquema general de la placa Análisis de Circuitos**

El módulo permite una gran variedad de configuraciones, por lo que para el desarrollo del manual se ha escogido una configuración Divisor de Tensión. En la Figura 3.3-3 se muestra el esquemático de la configuración utilizada:



**Figura 3.3-3: Esquema del divisor de tensión**

En esta configuración el alumno puede elegir entre los siguientes parámetros configurables:

- **Alimentación:** generador de funciones, fuente de alimentación o abierto.
- **Impedancias:** puede configurar el valor de  $Z1$ ,  $Z2$  y  $Z3$  pudiendo elegir entre resistencias, condensadores, bobinas, cortocircuito o circuito abierto. Las impedancias pueden ir tanto en serie como en paralelo.
- **Medidas de tensión:** 3 puntos de medida disponible ( $V1$ ,  $V2$ ,  $V3$ ) tanto para el multímetro como para el osciloscopio.
- **Medidas de resistencia:** 1 punto de medida ( $V1$ ).
- **Medidas de corriente:** 3 puntos a medir con el amperímetro ( $A1$ ,  $A2$ ,  $A3$ ).

### Bloque de Dispositivos

El bloque dispositivos está constituido por los instrumentos de medida, osciloscopio y multímetro, y de excitación, fuente de alimentación y generador de señal, de los cuales se utilizarán la mayoría de sus funcionalidades. El servidor laboratorio dispone de una tarjeta PCI-GPIB a demás de puertos USB para el control de los dispositivos y el experimento (ver Figura 3.3-1).

## 3.4 Aplicación software

La aplicación software necesaria para el control y monitorización del sistema, debe cumplir los siguientes aspectos:

- Se debe adaptar lo mejor posible a las especificaciones generales (ver apartado 3.2 *Especificaciones generales*).
- Se debe implementar sin excesiva complejidad, siendo fácilmente ampliable o modificable.

En el apartado 2.5 *Estructura y componentes de un laboratorio remoto*, se nombraron algunas de las posibilidades, a nivel software, para desarrollar las aplicaciones que se deberían ejecutar tanto en el ordenador cliente como en el servidor de un laboratorio remoto. Ahora se aborda con más detalle el diseño de la aplicación software.

- **Aplicación Cliente**

Esta aplicación es la que utilizaría el usuario remoto para desarrollar las diferentes prácticas disponibles. Teniendo en cuenta las funciones que tendría que realizar, las opciones que se han analizado son las siguientes:

**1) Aplicación específica** desarrollada utilizando un determinado lenguaje o entorno de programación (C, C++, Java, Visual Basic, .NET, Python, LabView, etc.) que debe ser instalada en el ordenador cliente. Esta solución presenta una gran versatilidad a la hora de diseñar la aplicación a cambio de una elevada dependencia del programador en cuanto a desarrollo, mantenimiento y escalabilidad.

**2) Aplicación Web** desarrollada utilizando lenguajes como HTML, Java script y XML, entre otros, que permiten aprovechar los servicios de Internet disponibles. Esta solución también ofrece como ventaja la versatilidad en el diseño de la aplicación, pero además incorpora que el estudiante puede acceder al servidor del laboratorio



mediante un explorador Web sin necesidad de tener instalado en su ordenador ningún programa.

**3) Aplicaciones** diseñadas para ceder el control del servidor del laboratorio al cliente, permitiendo manejar directamente el programa que se esté ejecutando en dicho servidor. Ejemplos de este tipo de aplicaciones son el Escritorio remoto de Windows o aplicaciones del tipo VNC (Virtual Network Computing). Estas soluciones son ventajosas ya que no es necesario programar aplicación alguna, pero si tienen la necesidad de tener instaladas estas aplicaciones en el ordenador cliente y establecer las medidas de seguridad necesarias para evitar accesos indeseados a recursos del servidor del laboratorio.

**4) Utilidades** disponibles en paquetes de software profesionales como Matlab (Matlab Web server) o LabView (Remote Panel) que permiten la interacción con el servidor del laboratorio con un reducido o nulo esfuerzo de programación. Estas soluciones tienen la ventaja de basarse en software de uso extendido en el ámbito universitario, teniendo en cuenta que tienen una escasa versatilidad y costes de las licencias de los paquetes profesionales.

**5) Visor 3D** para mundos virtuales. Es una aplicación capaz de visualizar un mundo o entorno virtual, en el cual los usuarios pasan a denominarse “avatares”. Los avatares pueden así explorar el mundo virtual, interactuar con otros avatares, establecer relaciones sociales, participar en diversas actividades tanto individuales como en grupo, y usar objetos o bienes virtuales.

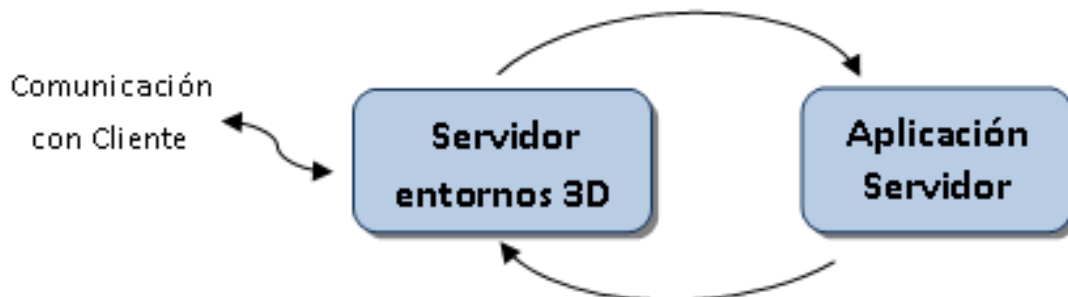
Teniendo en cuenta estas posibilidades, la opción escogida es el Visor 3D. La principal razón de escoger como aplicación para el usuario un visor 3D es el aumento de realidad a la hora de trabajar con el experimento remotamente. Pasar de una interfaz en 2D, siendo lo habitual el uso de una aplicación específica o un aplicación Web, a un entorno 3D, da un salto cualitativo a la hora de familiarizarte con el manejo de los instrumentos. Por ejemplo, al utilizar un entorno 3D, se puede realizar el conexionado de cables y componentes a la placa de pruebas igual que se realizaría en el laboratorio real.

Conviene aclarar que dicho visor sólo será la Interfaz de usuario, la representación gráfica y herramientas de 3D del mundo virtual que se ejecuta en el cliente, por lo que es necesario la utilización de un servidor que simule dicho mundo.

En el apartado 4.2.2 *Listado de visores*, se expondrán algunos de los visores que nos podemos encontrar actualmente y que se pueden utilizar con el servidor de mundos virtuales escogido (Opensim). También se escogerán algunos de ellos y se explicará paso a paso su instalación en el *ANEXO I*.

- **Sistema Software del Servidor**

En esta apartado tenemos que diferenciar dos aplicaciones que se ejecutarán en el servidor (Figura 3.4-1):



**Figura 3.4-1: Sistema software del servidor**

- **Servidor del entorno 3D.** Este servidor surge al utilizar como aplicación cliente un visor de entornos virtuales. Es la aplicación encargada de gestionar el mundo virtual y es accesible desde diferentes tipos de clientes o protocolos. En dicho entorno cada usuario dispondrá de un Avatar con el que podrá interaccionar con los diferentes objetos y otros usuarios dentro del entorno creado.

Para implementar esta aplicación se han analizado las siguientes opciones:

**1) Servidor SecondLife (SL):** es un metaverso<sup>1</sup> lanzado el 23 de junio de 2003, desarrollado por Linden Lab. El mundo de SL es, por tanto, un entorno privado, en el que se pueden comprar regiones virtuales en la que desarrollar el metaverso. Dicha región estaría alojada en los servidores propiedad de Linden Labs, por lo que no es posible su instalación o creación de mundos paralelos fuera de la región adquirida. El servicio que proporciona Linden Labs es anual, por lo que para mantener las regiones adquiridas y su contenido supone abonar su coste cada año.

El acceso a SL es gratuito, sólo es necesario crear una cuenta y tener acceso a internet. Sus usuarios, conocidos como avatares, pueden acceder a SL mediante el uso de uno de los múltiples programas de interfaz llamados visores. El avatar puede ser totalmente fiel a la persona real o una imagen completamente diferente. Los cambios básicos del avatar (complexión, forma del cuerpo, colores...), son bastante sencillos de realizar, para el resto, se necesitan algunos conocimientos de diseño en 3D.

**2) Servidor Opensim.** Opensim es una plataforma de código abierto que utiliza el mismo estándar que SecondLife para crear y gestionar mundos virtuales desde una infraestructura independiente a la de Linden Lab.

En cuanto a las ventajas que nos ofrece esta plataforma, tenemos que destacar dos principalmente. La primera es que se rompe definitivamente la dependencia con Linden Labs para poseer un servidor de mundos virtuales propio y sin ningún coste. Además, al romperse el monopolio en la extensión de territorio virtual, los costes bajan drásticamente, derribando una de las principales barreras de entrada. Y la segunda es que al ser código abierto nos permite ir desarrollando nuevas funcionalidades que se adapten a las necesidades de cada organización.

---

<sup>1</sup> Un metaverso es un mundo virtual ficticio. Los metaversos son entornos donde los humanos interactúan social y económicamente como iconos, a través de un soporte lógico en un ciberespacio que actúa como una metáfora del mundo real, pero sin las limitaciones físicas allí impuestas.

Como desventajas, el proyecto de Opensim aun no alcanza un estado de maduración tal para ser considerado igual SecondLife. Se está utilizando una versión en desarrollo (0.7.3), no totalmente estable, por lo que se pueden encontrar algunos problemas durante su uso. A diferencia de SecondLife, no se comparte la base de usuarios ni tampoco existe una economía interna, pues al ser una comunidad independiente, cada organismo controla servidor de forma diferente.

Al comparar estas dos opciones, claramente la utilización del servidor Opensim se ajusta más a nuestras necesidades. La principal razón de su elección es que el código abierto nos permite desarrollar nuevas funcionalidades según surjan nuevas necesidades, por lo que el que campo que podemos abarcar con él es mucho mayor. Otra de las ventajas que ofrece el código libre es el coste, ya que de haber desarrollado el mundo virtual en SecondLife supondría un gasto anual importante.

A diferencia de SecondLife, Opensim se instala en nuestro propio servidor (ordenador), o servidores funcionando en paralelo (Linux o Windows), lo que nos permite escalar la capacidad del mismo a nuestras necesidades. El tener el control del servidor también nos permite evitarnos los problemas de confidencialidad asociados a funcionar en un servidor controlado por Linden Labs y la fiabilidad de nuestro servicio depende únicamente de nosotros, dejando de depender de las incidencias incontrolables de Linden Labs.

- **Servidor laboratorio:** se encarga de la gestión de usuarios, controlar los instrumentos de medida (osciloscopio y multímetro) y excitación (fuente de alimentación y generador de señal) y el experimento, donde se implementan los diferentes circuitos electrónicos, así como de gestionar las comunicaciones HTTP con el servidor de entorno 3D, Opensim.

Para implementar esta aplicación se han analizado las siguientes opciones:

1) Una aplicación basada en el uso de paquetes software profesionales como LabView, VEE y MATLAB/Simulink.

2) Una aplicación basada en el uso de lenguajes de programación de propósito general como C, C++, Python, etc. Esta solución no requiere realizar gastos de compra en licencias de uso de software pero su principal inconveniente radica en que el esfuerzo en programación es muy elevado.

Entre las diferentes opciones de herramientas de programación, con sus ventajas e inconvenientes, está la utilización del paquete software Labview<sup>2</sup>. Analizadas las prestaciones que ofrece esta herramienta de programación y consideradas de nuevo las especificaciones que se han impuesto para desarrollar esta versión de laboratorio remoto, se ha decidido que su utilización resulta apropiada para implementar la aplicación servidor, que se comunicaría con el usuario remoto y controlaría la arquitectura hardware propuesta.

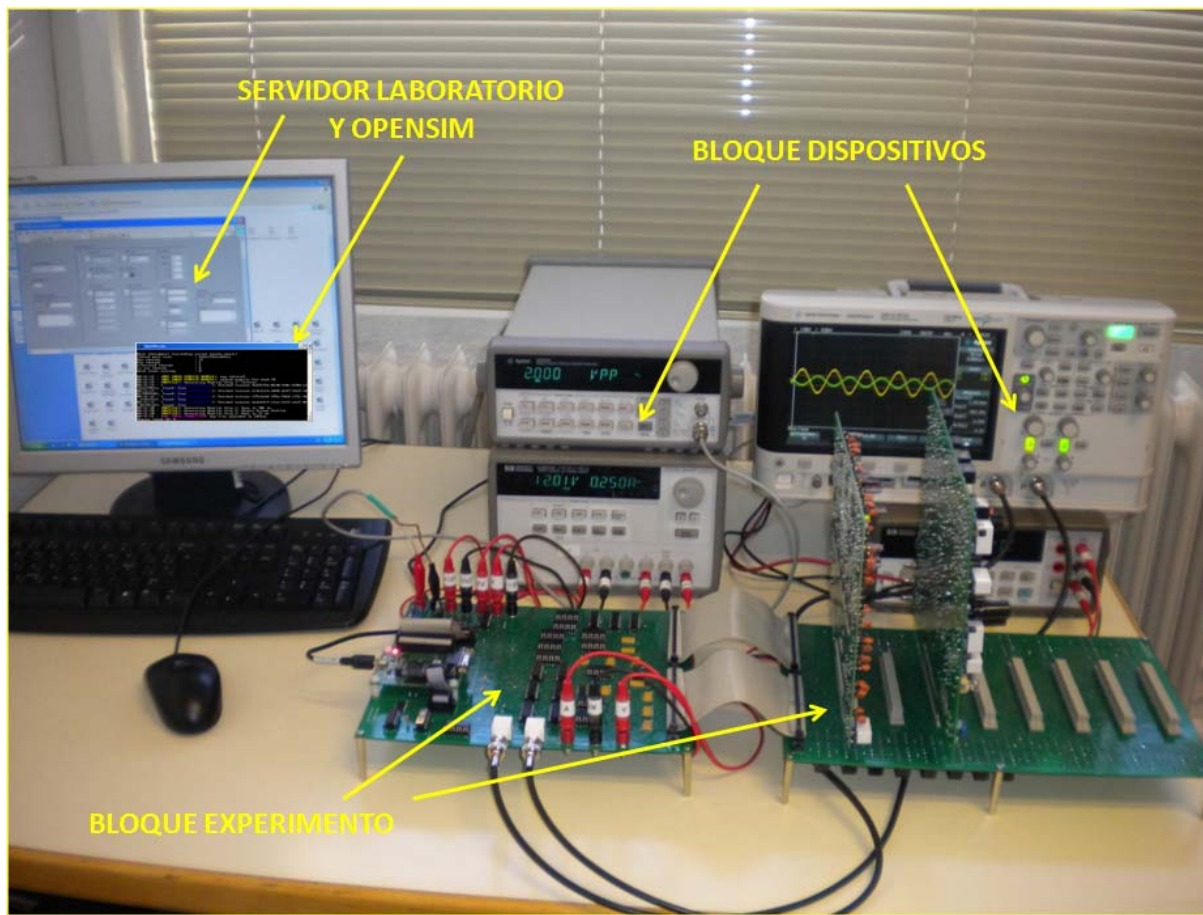
## 3.5 Sistema final: Laboratorio remoto de electrónica analógica

En la Figura 3.5-1 se muestra una imagen real de la plataforma desarrollada que actualmente está ubicada en un laboratorio del Departamento de Sistemas Electrónicos y de Control (EUIT Telecomunicación) de la Universidad Politécnica de Madrid.

En la figura se puede apreciar un ordenador, que es el encargado de ejecutar la aplicación de entornos virtuales Opensim, y la aplicación servidor laboratorio. También podemos observar los instrumentos utilizados (bloque dispositivos en la Figura 3.3-1), que junto con el sistema hardware (bloque experimento en la Figura 3.3-1) conectados a dicho ordenador mediante los buses de comunicación GPIB y USB.

---

<sup>2</sup> LabVIEW es una herramienta creada por National Instruments para medición, pruebas, control y automatización mediante programación gráfica.



**Figura 3.5-1: Sistema completo del servidor**

# **4**

## **DESARROLLO DEL SISTEMA SOFTWARE**



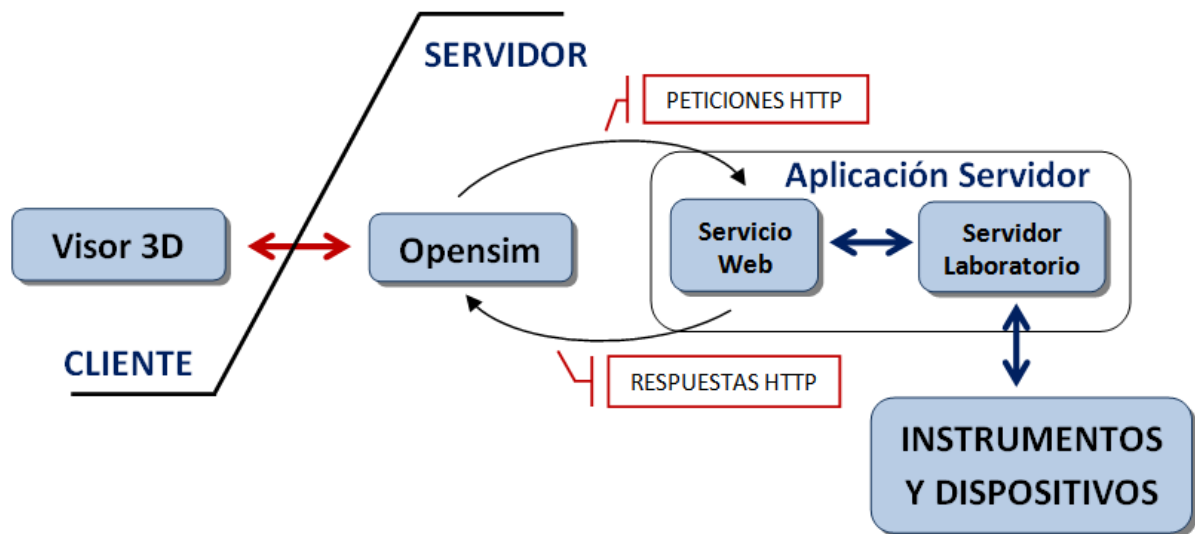


## 4.1 Introducción

Los aspectos clave que se han tenido en cuenta para el desarrollo de la aplicación son los siguientes:

- La estructura se debe adaptar lo mejor posible a las especificaciones descritas en el apartado 3.2 *Especificaciones generales*
- Su implementación debe estar orientada a una fácil depuración

En el apartado 3 *SOLUCIÓN ESCOGIDA PARA EL LABORATORIO REMOTO*, se indicaron las opciones elegidas, para desarrollar las aplicaciones que se deberían ejecutar en el ordenador cliente y en el servidor del laboratorio remoto. El sistema software completo quedaría representado como se muestra en la Figura 4.1-1:

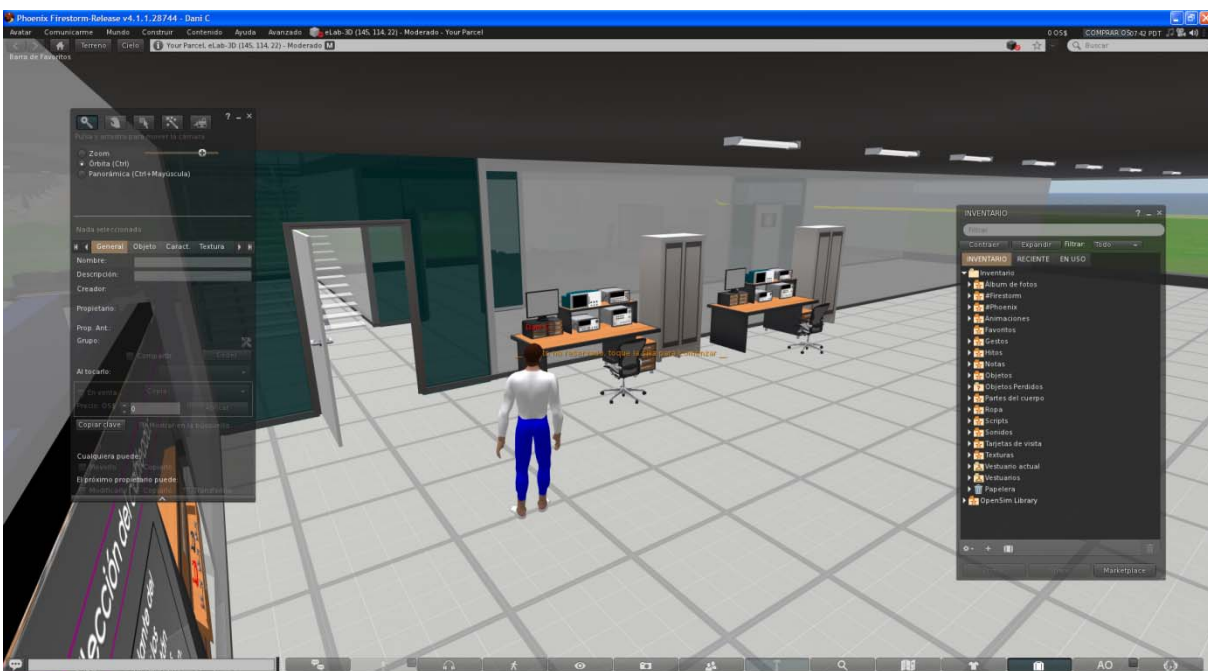


**Figura 4.1-1: Sistema software completo**

Ahora se van a detallar las dos partes diferenciadas del sistema, explicando más en profundidad las soluciones escogidas para la aplicación cliente y servidor, y analizando cada una de sus partes.

## 4.2 Aplicación cliente: Visor 3D

Después de analizar las diferentes posibilidades en el apartado 3.4 *Aplicación software*, la solución escogida fue utilizar un entorno 3D mediante un visor, que utilizará el usuario para el acceso remoto al laboratorio y así poder desarrollar las diferentes prácticas disponibles. Ya se comentó que dicho visor solo será la interfaz de usuario, la representación gráfica del mundo virtual que visualiza el cliente. En la Figura 4.2-1 se muestra una imagen del visor:



**Figura 4.2-1: Visor del entorno 3D**

Para aclarar el concepto de visor, se puede hacer una analogía entre el visor y una web de internet. Las webs se alojan en servidores y mediante un cliente (explorador), podemos visualizar su contenido.

Ocurre exactamente lo mismo en el mundo virtual. El mundo virtual está alojado en un servidor (o servidores), para poder gestionar tu avatar y ver lo que ocurre dentro, necesitamos un cliente. Igual que en Internet existen diferentes exploradores, existen diversos visores de diferentes empresas. Cada uno suele tener sus propiedades y detalles, aunque a grandes rasgos, todos hacen lo mismo.

Conviene aclarar aquí que el visor es sólo una herramienta para poder utilizar el laboratorio y no es necesario ser un experto en su manejo para sacar provecho de las prestaciones del Laboratorio de Electrónica. También hay que hacer hincapié en que se está trabajando con versiones de software libre en desarrollo y que es muy probable que al utilizar el visor haya algunas opciones que no funcionen correctamente. Este problema se puede solucionar en muchas ocasiones haciendo uso de otro visor, teniendo en cuenta que el manejo de todos ellos es similar.

Antes de abordar el análisis de algunos de los diferentes visores existentes actualmente, se va a explicar el concepto de entorno virtual (o mundo virtual) y algunas de sus características que han hecho que esta sea la opción escogida como interface de usuario para el acceso al laboratorio remoto.

También se hará referencia a Second Life (apartado 4.2.1.3 *Second Life*), ya que es el mundo virtual social más utilizado en la actualidad y de donde nace el software que se utilizará en este laboratorio remoto.

## 4.2.1 Entorno virtual

### 4.2.1.1 Conceptos

Un entorno virtual es una representación gráfica de un modelo del mundo real que se consigue mediante un proceso de renderizado<sup>3</sup>, donde se genera una imagen 2D a partir de una en 3D, a través de la cual los usuarios pueden interactuar entre sí mediante personajes virtuales llamados avatares, así como utilizar y crear objetos. Los mundos virtuales están destinados a que sus usuarios los habiten e interactúen.

Una de las características de los mundos virtuales es que estos deben simular un entorno en tres dimensiones en un medio de dos dimensiones: la pantalla. Este hecho, añadido a la menor naturalidad para moverse y cambiar el enfoque de lo que se está viendo dentro de los mundos virtuales, hace necesario un diseño específico

---

<sup>3</sup> Renderizar: es un término usado para referirse al proceso de generar una imagen desde un modelo.

de la interacción dentro del mundo virtual y de todas las aplicaciones dentro del mismo, enfocado directamente en la experiencia del usuario.

Existen principalmente dos razones por las que se utilizaría un mundo virtual: porque es útil o porque es atractivo. En el segundo caso, que corresponde a la gran mayoría, la experiencia del usuario es un condicionante principal del éxito del mundo virtual, puesto que el usuario no está obligado a utilizar un cierto mundo virtual en particular, por lo que si la experiencia en uno de ellos no le agrada, simplemente dejaría de usarlo.

La usabilidad se considera como uno de los atributos básicos de cualquier sistema interactivo, y corresponde a una medida que dimensiona que tan fácil es para los usuarios utilizar el sistema para completar su objetivo.

Los mundos virtuales aportan una valiosa característica adicional, un mayor nivel de inmersión en el aprendizaje, ya que genera en los estudiantes mayor sensación de presencialidad, gracias a la libertad y amplitud de movimiento, los mayores detalles al visualizar y la escala a la que se proyectan los ambientes. Todos estos beneficios garantizan la formación integral de un estudiante, tal como se ha dado en la educación tradicional, he aquí la importancia de su implementación como una nueva herramienta de aprendizaje.

#### **4.2.1.2**    *Características generales*

Una vez establecida la definición del concepto de mundo virtual, es necesario detallar cada una de las características necesarias para catalogar a una aplicación como un mundo virtual.

Las condiciones que debe cumplir una aplicación para ser catalogada como un mundo virtual son las siguientes:

1. Ambiente de aprendizaje multisensorial: un usuario accede al programa a través de una interfaz que simula un entorno, lo que le proporciona una sensación de presencialidad
2. El mundo y todos sus habitantes están sometido a un conjunto de reglas “físicas” irrompibles.
3. El usuario es representado por un solo individuo en el mundo, conocido como avatar.
4. El mundo es compartido, participan muchos usuarios simultáneamente, unificando los conocimientos y criterios en base a su participación activa.
5. Los usuarios deben poder interactuar y comunicarse entre sí y deben poder interactuar con el mundo, creando, modificando y poseyendo contenidos virtuales.
6. La interacción con el mundo y otros usuarios es en tiempo real.
7. El mundo es (parcialmente) persistente, el mundo existe siempre y recuerda la localización de personas y objetos, estén o no presentes los usuarios que lo han creado.

Todas estas condiciones, exceptuando la primera, que es auto explicativa y no requiere mayores explicaciones, serán revisadas en detalle en los puntos siguientes.

## AVATARES

Un avatar es un personaje o habitante del mundo virtual que representa y es controlado por el usuario. Toda interacción con el mundo virtual y con los demás usuarios se lleva a cabo a través del avatar. Un avatar tiene al menos las capacidades de moverse, comunicarse con otros avatares y utilizar objetos del entorno.

El avatar es una de las características distintivas y más importantes de los mundos virtuales, pues representa el hecho que un usuario es caracterizado por un único personaje dentro del mundo virtual que lo identifica de los demás usuarios. En general, el mundo virtual debe permitir al usuario un cierto grado de personalización de su avatar, con el objetivo de distinguirlo de los demás avatares dentro del mundo virtual.

## LEYES FÍSICAS

Las leyes físicas son aquellas que rigen y delimitan el accionar de los avatares y objetos dentro del mundo. Estas reglas son propias a cada mundo virtual, no pueden ser alteradas por los usuarios y su objetivo es el correcto funcionamiento del mundo virtual. Como ejemplo de esto tenemos implementaciones de la fuerza de gravedad, la imposibilidad de atravesar paredes o el suelo mismo, limitaciones al momento de crear contenido dentro del mundo virtual (tamaño máximo, formas y colores permitidos o disponibles), etc.

## MUNDO COMPARTIDO

Un mundo virtual siempre es compartido, es decir, debe tener la capacidad de tener a más de un usuario al mismo tiempo. Esto también implica que si dos usuarios están en un mismo lugar dentro del mundo virtual, deben ver lo mismo (sin contar los detalles de configuración grafica o de efectos especiales que tenga cada usuario) incluidos los avatares de los demás usuarios. Cualquier cambio que un usuario realice al mundo virtual debe ser visible por todos los demás usuarios y debe afectarlos a todos.

## INTERACCIÓN Y COMUNICACIÓN

El usuario, a través de su avatar debe ser capaz de interactuar con elementos del mundo virtual, ya sea creando objetos o utilizando herramientas u objetos.

En cuanto a la comunicación, el mundo virtual debe dar a los usuarios la capacidad de comunicarse entre sí a través de sus avatares, ya sea a través de texto, voz, sonidos, gestos o material audiovisual.

## INTERACCIÓN EN TIEMPO REAL

Toda interacción dentro de un mundo virtual, ya sea entre avatares o de un avatar con el mundo en sí, debe ser en tiempo real, es decir, debe generar un resultado inmediato. Por ejemplo, si un avatar recoge un objeto, este debe ser accesible

inmediatamente por el usuario. Todo evento y toda acción llevada a cabo dentro de un mundo virtual debe ser visible para todos los usuarios de forma simultánea e inmediata.

### MUNDO PERSISTENTE

Esta última condición, implica que tanto el mundo virtual mismo, como los cambios que ocurren en él, ya sea por el motivo que sea, se mantienen en el tiempo.

#### *4.2.1.3 Second Life*

Ya que parte del sistema software mencionado en el apartado *4.1 Introducción*, en concreto el cliente (visor 3D) y el servidor del mundo virtual (Opensim), nacen de **Second Life**, es necesario hacer referencia a él.

El metaverso denominado Second Life (SL), es un mundo virtual 3D creado por Linden Lab y fundado por Philip Rosedale. Es un mundo que está distribuido en una amplia red de servidores y que puede ser accedido a través de Internet. Este programa proporciona a sus usuarios o “residentes” herramientas para explorar el mundo virtual, interactuar con otros residentes, establecer relaciones sociales, participar en diversas actividades tanto individuales como en grupo y crear y comerciar propiedad virtual y servicios entre ellos. Second Life es el mundo virtual social más exitoso y con mayor número de usuarios del momento, así como uno de los más explotados comercialmente.

Second Life se destaca del resto de los mundos virtuales 3D por el gran número de recursos que se van incluyendo en su desarrollo, lo que le hacen destacarse en su potencial aplicativo. Es sin duda el que ofrece una mayor sensación de inmersión para el usuario.

En definitiva, lo que se ha hecho en el laboratorio remoto es utilizar la tecnología surgida a partir de Second Life, para crear y gestionar el mundo virtual donde está alojado el laboratorio. Esto se consigue sustituyendo los servidores propiedad de

Linden Lab por uno de gestión propia como Opensim (explicado en profundidad en el apartado 4.3.1 *Opensim*), y para tener acceso a dicho servidor, se utiliza uno de los visores de entornos 3D surgidos del creado para Second Life (llamado Second Life viewer). Tanto el servidor como el visor son de software libre.

## 4.2.2 Listado de visores

Una vez que se han comentado los aspectos básicos de los entornos o mundos virtuales, es hora de analizar las aplicaciones de que disponemos para acceder al mundo virtual. Detallamos aquí algunos de los visores que hay en la actualidad y que podemos utilizar para conectarnos al laboratorio remoto creado.

No existe ninguna ventaja entre los usuarios que usen uno u otro, el visor que se escoja debe ser estable y que tenga una interfaz simple y agradable para cada usuario final. Todos los visores aquí nombrados son compatibles con Opensim y SecondLife, y son soportados tanto en Windows como en Linux y MAC OSX.

### Visor oficial de Second Life

URL: <http://secondlife.com/?lang=es-ES>

DESCARGA: <http://secondlife.com/support/downloads/?lang=es-ES>

**Descripción:** Visor Oficial para Second Life, desarrollado y distribuido por Linden Lab. Su código es la base para el resto de visores desarrollados. Actualmente se está utilizando el código perteneciente al visor 2.

---

### Visor Phoenix

URL: <http://www.phoenixviewer.com/>

DESCARGA: <http://www.phoenixviewer.com/downloads.php>

**Descripción:** El Visor Phoenix es un visor creado por el Equipo de Desarrollo del Proyect Phoenix Visor Inc. Cuenta con gran conjunto de funcionalidades y un equipo de soporte técnico disponible 24/7.

---



## Visor FireStorm



URL: <http://www.phoenixviewer.com/>

DESCARGA: <http://www.phoenixviewer.com/downloads.php>

**Descripción:** es la nueva generación de visor del Proyecto Phoenix. Firestorm está basado en el código del visor 2 de Second Life, con muchos cambios y funcionalidades añadidos.

---

## Visor Imprudence



URL: [http://wiki.kokuaviewer.org/wiki/Main\\_Page](http://wiki.kokuaviewer.org/wiki/Main_Page)

DESCARGA: <http://wiki.kokuaviewer.org/wiki/Downloads>

**Descripción:** Imprudence busca mejorar la interface de usuario y un manejo más sencillo del visor a través de la participación de su comunidad de usuarios.

---

## Visor Singularity



URL: <https://sites.google.com/site/singularityviewer/>

DESCARGA: <http://www.singularityviewer.org/downloads>

**Descripción:** proviene del visor abandonado Ascent, actualizándolo desde la base para una mayor estabilidad, también provisto de compatibilidad con las nuevas funciones de Second Life.

---

Cabe destacar que los visores listados aquí no son todos los visores disponibles, sino que se ha optado por listar los visores mas utilizados en mundos virtuales basados en Second Life u Opensim.

En el apartado 5 MANUAL DE USUARIO I: VISOR 3D, se pueden consultar los aspectos básicos para el manejo del visor, que se ha elegido para operar con el laboratorio de electrónica.

También en el apartado 9 *ANEXO I*, se pueden consultar las instrucciones para poder instalar alguno de los visores aquí descritos.

### 4.2.3 Requisitos para la instalación del visor

Para ejecutar correctamente los visores se debe disponer de un ordenador con algunos requisitos. Los ordenadores portátiles, incluso recientes, equipados con tarjetas gráficas integradas no permitirán utilizar el software con comodidad. Para explotar al máximo los recursos gráficos disponibles, se necesitará una tarjeta gráfica con al menos 512 MB (DDR3 y DDR4) y 2 GB de Ram.

En la Tabla 4.2-1: Comparativa entre sistemas operativos se muestran los requisitos mínimos para el correcto funcionamiento del visor, dependiendo del sistema operativo en el que ejecute:

| Windows   | Mac OS X  | Linux  |
|---|---|--|
| <b>Conexión a internet:</b> Cable o DSL<br><br><b>Sistema operativo:</b> XP, Vista o Windows 7<br><br><b>Procesador:</b> 800 MHz Pentium III o Athlon, o superior<br><br><b>Memoria RAM:</b> 512MB o más.<br><br><b>Resolución de pantalla:</b> 1024x768 pixeles o superior<br><br><b>Tarjetas gráficas soportadas:</b> <ul style="list-style-type: none"> <li>- NVIDIA GeForce 6600 o superior</li> <li>- ATI Radeon 9500 o superior</li> <li>- ATI Radeon 8500, 9250 o superior</li> <li>- Intel 945 chipset</li> </ul> | <b>Conexión a internet:</b> Cable o DSL<br><br><b>Sistema operativo:</b> Mac OS X 10.4.11 o posterior<br><br><b>Procesador:</b> 1.5 GHz Intel based Mac recomendado Intel Core 2 Duo a 2 GHz<br><br><b>Memoria RAM:</b> 512 MB o más<br><br><b>Resolución de pantalla:</b> 1024x768 pixeles o superior<br><br><b>Tarjetas gráficas soportadas:</b> <ul style="list-style-type: none"> <li>- ATI Radeon 9200 and above</li> <li>- NVIDIA GeForce 2, GeForce 4</li> </ul> | <b>Conexión a internet:</b> Cable o DSL<br><br><b>Sistema operativo:</b> Se requiere un entorno razonablemente moderno de 32-bit. Si está ejecutando una distribución Linux de 64-bit, entonces necesitará tener instalado un entorno de compatibilidad con 32 bits.<br><br><b>Procesador:</b> 800 MHz Pentium III, Athlon, a 1.5 GHz o superior<br><br><b>Memoria RAM:</b> 512MB o más<br><br><b>Resolución de pantalla:</b> 1024x768 pixeles o superior<br><br><b>Tarjetas gráficas soportadas:</b> <ul style="list-style-type: none"> <li>- NVIDIA GeForce 6600, o superior</li> <li>- OR ATI Radeon 8500, 9250 o superior</li> </ul> |

**Tabla 4.2-1: Comparativa entre sistemas operativos<sup>4</sup>**

<sup>4</sup> <http://secondlife.com/support/system-requirements/?lang=en-US>

## 4.3 Sistema software del servidor

Como se puede observar en la Figura 4.1-1, que muestra el sistema software completo, el servidor consta de dos partes diferenciadas:

- El servidor **Opensim**, encargado de la gestión del mundo virtual.
- La aplicación realizada con el lenguaje de programación LabView y que se encarga de la gestión del hardware, entre otras cosas. Ésta a su vez la podemos dividir también en dos partes bien diferenciadas:
  - **Servicio Web**, se encarga de la comunicación con el servidor del mundo virtual Opensim.
  - **Servidor de laboratorio**, es el encargado de la gestión de usuarios y de gobernar el sistema hardware.

### 4.3.1 Opensim

En enero de 2007, Linden Labs abrió el código del programa cliente (visor explicado anteriormente), para que fuera modificado y mejorado por la comunidad de desarrolladores independientes de SecondLife. Se esperaba entonces que a lo largo de ese año abriera también el código del programa servidor e incluso lo liberara, renunciando por tanto total o parcialmente al monopolio sobre el suelo del mundo virtual, pero no fue así.

Sin embargo, mediante ingeniería inversa y a partir del código del cliente, a partir de julio de 2007 grupos de voluntarios generaron el primer estándar libre (y compatible) de creación y gestión de mundos virtuales: **Opensim**. En la Figura 4.3-1 se muestra el logo de Opensim:



**Figura 4.3-1: Logo Opensim**

Por lo tanto, Opensim es un gestor de código abierto, para simular entornos virtuales, que utiliza el mismo estándar que Second Life para comunicarse con sus usuarios. Por esto hace posible que los usuarios puedan utilizar el mismo software que provee Linden Labs (visor).

Para ejecutar Opensim en modo autónomo, es necesario la utilización de bases de datos como SQLite o MySQL.

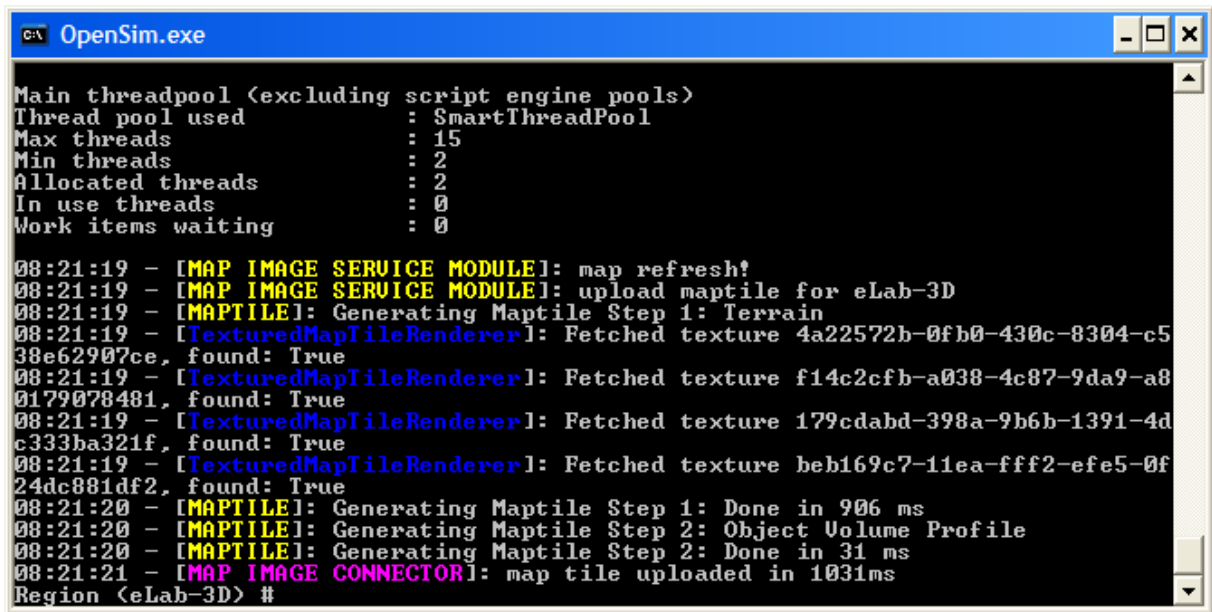
Por defecto, al instalar Opensim se ejecuta SQLite, por lo que no necesita de configuración por parte del usuario. En cambio si se quiere tener el control de la base de datos, poder acceder a la información o se necesita una base de datos extensa se debe utilizar MySQL. Para poder utilizar MySQL es necesaria una mínima configuración de los archivos opensim.ini, GridCommon.ini o Standalone.ini, según la configuración de Opensim<sup>5</sup>.

Por último, debe decirse que al ser de código abierto, su código fuente está disponible para cualquiera, puede ser descargado, modificado y utilizado libremente. Está escrito en la plataforma .NET y es multiplataforma, se puede ejecutar tanto en Windows con .NET Framework instalado como en Linux y Mac OS/X con Mono instalado.

En la Figura 4.3-2 se muestra la consola de comandos de Opensim, desde donde se gestiona y controla el mundo virtual:

---

<sup>5</sup> Referencia a los pasos a seguir para configurar la base de datos MySQL:  
<http://usvemo.wordpress.com/2011/01/26/230/>



```

OpenSim.exe
Main threadpool <excluding script engine pools>
Thread pool used      : SmartThreadPool
Max threads           : 15
Min threads           : 2
Allocated threads     : 2
In use threads        : 0
Work items waiting    : 0

08:21:19 - [MAP IMAGE SERVICE MODULE]: map refresh!
08:21:19 - [MAP IMAGE SERVICE MODULE]: upload maptile for eLab-3D
08:21:19 - [MAPTILE]: Generating Maptile Step 1: Terrain
08:21:19 - [TexturedMapTileRenderer]: Fetched texture 4a22572b-0fb0-430c-8304-c5
38e62907ce, found: True
08:21:19 - [TexturedMapTileRenderer]: Fetched texture f14c2cfb-a038-4c87-9da9-a8
0179070481, found: True
08:21:19 - [TexturedMapTileRenderer]: Fetched texture 179cdabd-398a-9b6b-1391-4d
c333ba321f, found: True
08:21:19 - [TexturedMapTileRenderer]: Fetched texture beb169c7-11ea-fff2-efe5-0f
24dc881df2, found: True
08:21:20 - [MAPTILE]: Generating Maptile Step 1: Done in 906 ms
08:21:20 - [MAPTILE]: Generating Maptile Step 2: Object Volume Profile
08:21:20 - [MAPTILE]: Generating Maptile Step 2: Done in 31 ms
08:21:21 - [MAP IMAGE CONNECTOR]: map tile uploaded in 1031ms
Region <eLab-3D> #

```

Figura 4.3-2: Consola de comandos Opensim

#### 4.3.1.1 *Mundos propietarios frente a mundos de código abierto: Second Life frente a Opensim*

En el caso de mundos inmersivos propietarios como Second Life, una serie de servidores pertenecientes a las empresas propietarias controlan el acceso a la plataforma, el envío de los datos referentes a la simulación del mundo en el ordenador del usuario, así como el almacenamiento de todas las configuraciones y datos asociados a los terrenos virtuales, a los avatares y a todos los contenidos creados por los mismos guardados en el inventario o en el terreno. Por tanto, las capacidades de los usuarios en términos de creación de contenidos y control sobre el mundo virtual están siempre supeditadas a la voluntad o fines comerciales de las empresas propietarias.

Sin embargo, el uso de Opensim permite a los usuarios implementar sus propios servidores en los que alojar el mundo virtual, definir las capacidades de los mismos y, en definitiva, tener control total sobre el contenido que crean y la forma de explotarlo. Algunas de sus ventajas en relación a las limitaciones de otras plataformas son:

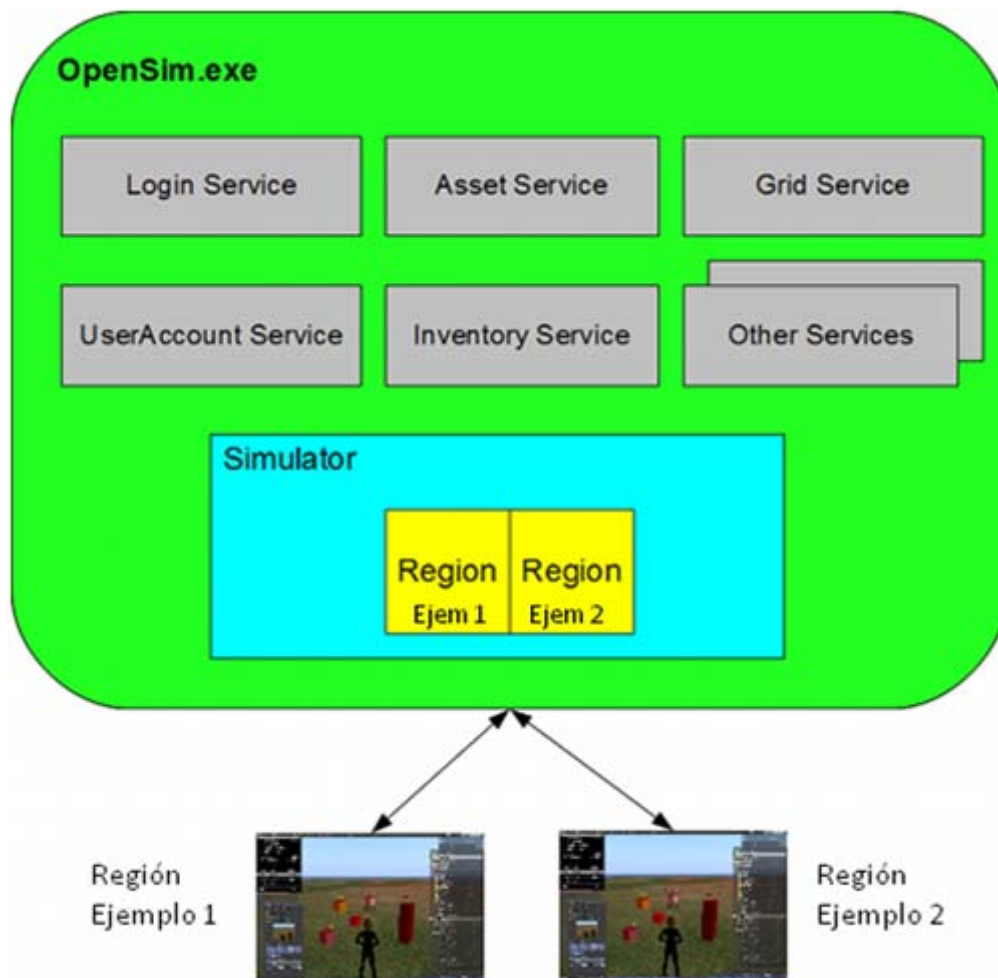
- Resulta una tecnología ideal para investigación y enseñanza, ya que es de código abierto, puede ser fácilmente modificado y ampliado, se puede instalar a nivel local con un coste mínimo posibilitando la posesión de un espacio virtual donde analizar, explorar y experimentar.
- Terreno virtual de cualquier extensión, libre y gratuito.
- Existen visores específicos que permiten características prohibidas en Second Life (por ejemplo poder hacer un backup a nuestro PC de nuestro inventario), ya que tenemos el control completo del servidor.
- Nos permite poder instalarlo en modo Standalone, es decir, en nuestro propio PC y sin necesidad de conectarlo a un Grid (conjunto de mundos virtuales).
- Obviamente, por lo dicho, también podemos crear nuestro propio grid con varios servidores y regiones y hacerlo privado para un grupo de gente. Esto es ideal para organizaciones e instituciones que buscan un entorno como el de Second Life pero no quieren utilizarlo por motivos varios (económicos, privacidad, control, etc.).
- Las prestaciones que soporta prácticamente son las mismas funcionalidades que Second Life (scripts, meshes, sculpts, windlight, etc.). Soporta la utilización de varios lenguajes, no solo LSL (el lenguaje de scripts de SecondLife), sino también OSL (Opensim Script language) y C#.
- Evitamos los problemas de confidencialidad asociados a funcionar en un servidor controlado por Linden Labs.
- La fiabilidad de nuestro servicio depende de nosotros. Dejamos de depender de las incidencias incontrolables de Linden Labs.

#### **4.3.1.2    *Arquitectura de Opensim***

Opensim se ejecuta como un proceso en el sistema, pero a su vez ese proceso está compuesto por muchos servicios (gestión de usuarios, inventario, regiones...) por lo que nos ofrecen la posibilidad de ejecutarlo de varios modos diferentes, de manera que podamos distribuir la carga de ese proceso entre varios servidores en caso de que sea necesario, los más utilizados son el modo Standalone y Grid.

### Modo Standalone

Es el modo por defecto del simulador. En este modo el proceso OpenSim.exe es independiente y gestiona todas las regiones y usuarios del simulador. El simulador puede tener todas las regiones que se quiera, pero todas son gestionadas por el mismo servidor. Este modo es útil para servidores pequeños o para realizar pruebas de configuración.

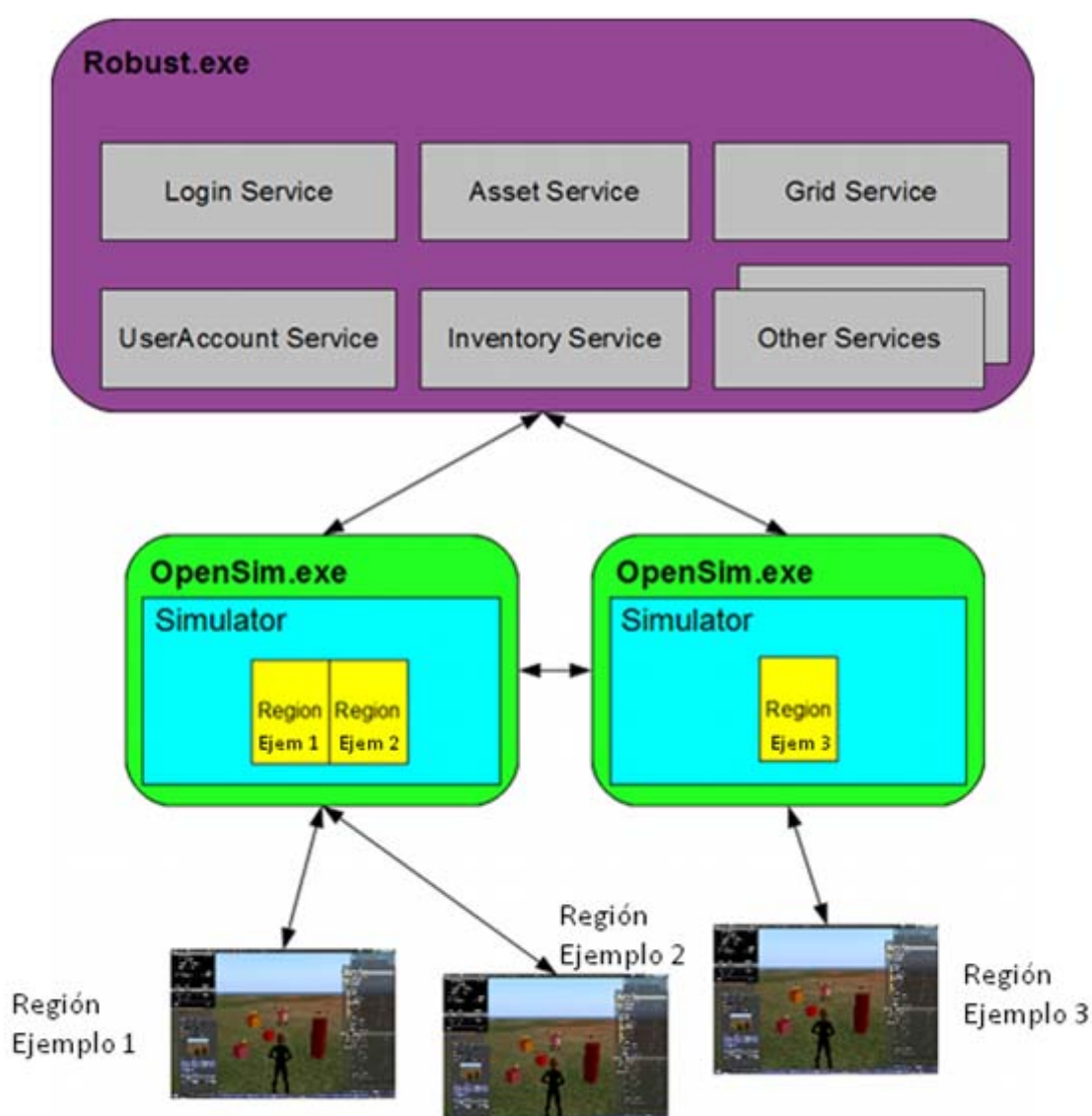


**Figura 4.3-3: Arquitectura Standalone**

Como se observa en la Figura 4.3-3, los servicios y las dos regiones de ejemplo son gestionados por el mismo proceso.

## Modo Grid

En este modo, el simulador está distribuido entre varios procesos, que pueden estar en varias máquinas distintas. El proceso Robust.exe es el que interconecta todos los demás procesos y se encarga de gestionar los servicios de usuarios, inventarios etc.



**Figura 4.3-4: Arquitectura Grid**

En la Figura 4.3-4, tenemos tres procesos: El Robust se encarga de gestionar los servicios e interconectar las tres regiones de ejemplo y dos procesos Opensim ejecutándose en servidores diferentes que se ocupan de la gestión de sus regiones.



### 4.3.1.3 Manejo del visor para la creación del mundo virtual

En el apartado 4.2 *Aplicación cliente*, se han comentado las prestaciones de los visores de cara al usuario del mundo virtual. Pero ésa no es su única función.

El visor es una parte importante del desarrollador, ya que permite la construcción dentro del mundo virtual, modificar la apariencia de objetos y avatares, crear los programas o scripts, utilizando el lenguaje de programación LSL (Linden Scripting Language). En definitiva, nos permite **crear el mundo virtual** y darle funcionalidad.

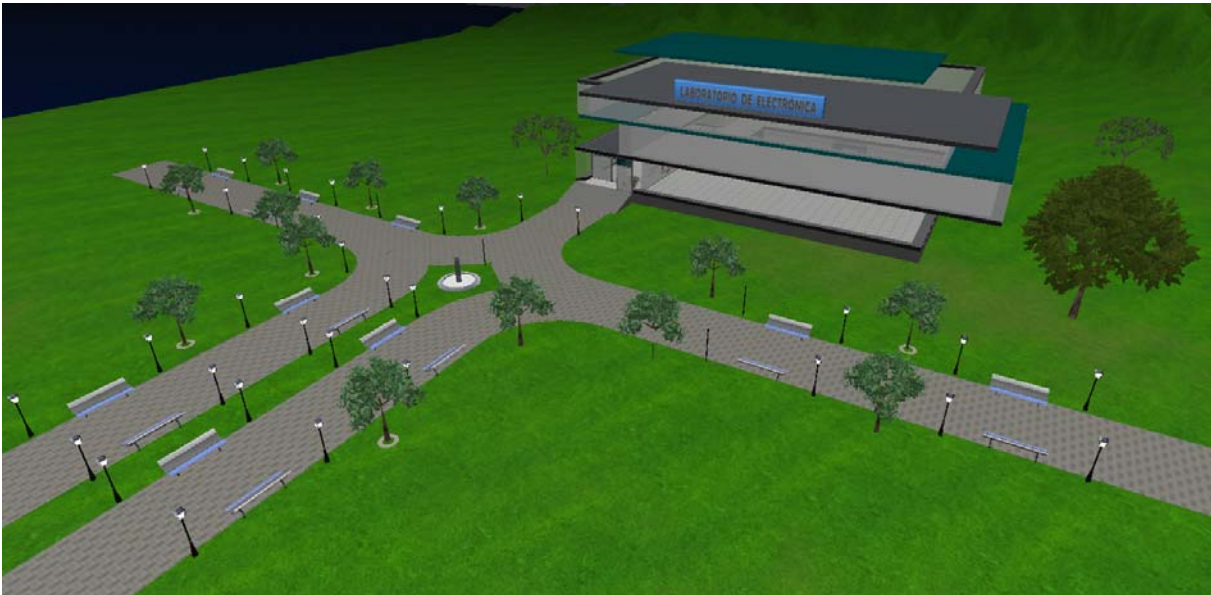
A continuación se proporcionan algunas indicaciones básicas para facilitar el manejo del visor de cara a la creación del mundo virtual.

#### Región o Isla

La región o terreno donde se aloja el mundo virtual coloquialmente se llama 'Isla'. Son espacios de unos 65.000 m<sup>2</sup> virtuales donde se creará todo lo relacionado al mundo virtual. Como se ha mencionado anteriormente en la arquitectura del servidor, podemos encontrarnos varias islas en un único servidor o enlazar varias islas de diferentes servidores, consiguiendo así un terreno de mayor extensión. En las Figura 4.3-5 se muestra la isla inicial y en la Figura 4.3-6 una isla con contenido, (se muestra el edificio del laboratorio de electrónica):



**Figura 4.3-5: Isla inicial del mundo virtual**



**Figura 4.3-6: Ejemplo de isla ya construida**

## Objetos

Los objetos se crean, editan y eliminan desde el visor. En Opensim son llamados prims o primitivas, que son formas básicas como: cubos, cilindros, conos...

Cada objeto tiene un inventario en el que puede haber scripts, texturas, sonidos, otros objetos, etc. Los scripts sólo pueden acceder a los elementos de su propio inventario. También puede haber objetos complejos, que se crean enlazando varias primitivas.

Cada región de terreno puede soportar un número limitado de objetos o prims. El número de prims de una región está directamente relacionada con su tamaño, así la región más grande es capaz de soportar 15000 prims. Un aspecto importante a tener en cuenta es que cuanto mayor sea el número de primitivas que haya en la región, más carga debe soportar el servidor Opensim, llegando a ralentizar bastante el sistema.

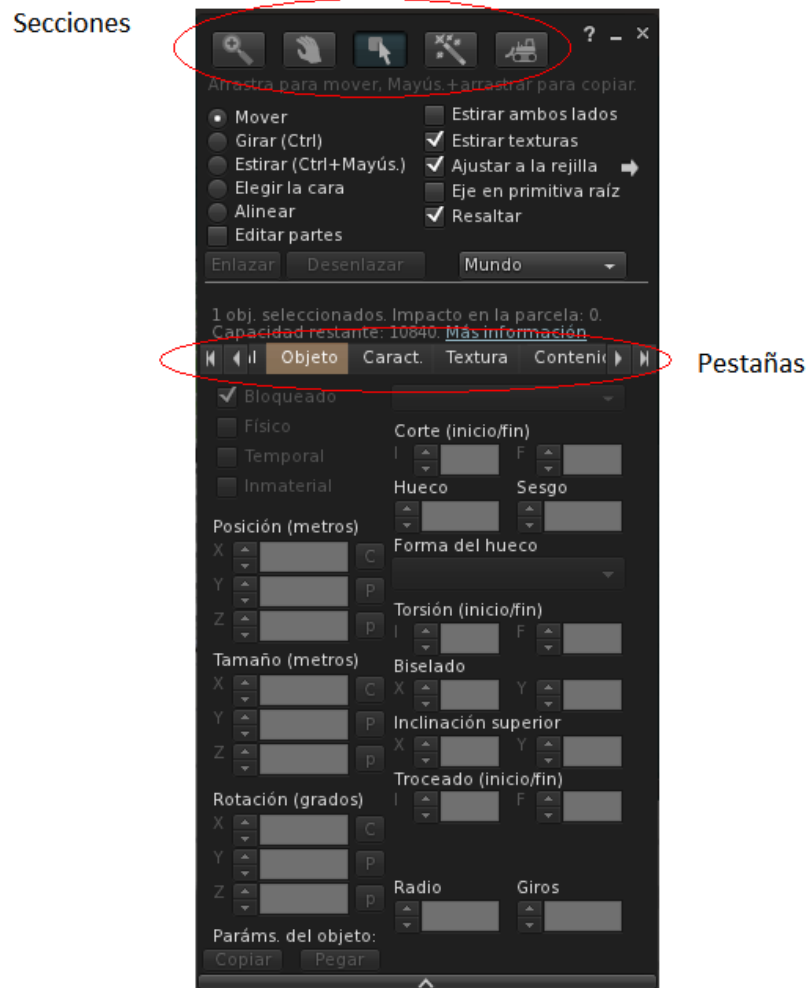
Una solución que se ha adoptado para que el laboratorio tenga el menor número de primitivas es crear objetos dinámicamente según sean necesarios. Por ejemplo, en cada puesto de laboratorio hay cajones que contienen los cables de los

instrumentos, las placas de prácticas y los componentes electrónicos, por lo que inicialmente al estar dentro de sus cajones, no están creados en el mundo virtual. En el momento en el que un avatar abre un cajón, los objetos se crean. Si en el desarrollo de una práctica un alumno no necesita todos los cables o todos los componentes, éstos no se crean y no cargan el sistema innecesariamente.

Esta solución ha sido clave en el desarrollo del laboratorio, ya que minimiza en gran medida la carga del servidor.

### Construcción y edición de objetos

En la Figura 4.3-7 se muestra la ventana de construcción del visor 3D. En la parte superior de la figura disponemos de una serie de pestañas:



**Figura 4.3-7: Ventana de construcción**

- General: Información del objeto (nombre, dueño...) y permisos del mismo.
- Objeto: Propiedades del objeto como tamaño, posición, rotación. Aquí también podemos bloquear el objeto o hacer que se pueda atravesar, entre otras cosas.
- Características: Permite dar flexibilidad o luz al objeto.
- Textura: Cambiar la imagen que se muestra en las caras del objeto, color, transparencia, brillo, resplandor...
- Contenido: Aquí es dónde se encuentra el inventario del objeto en el que se alojan los scripts, texturas, sonidos...

Y de secciones:

- Visión: Acercar o alejar la cámara con respecto al objeto.
- Mover: Mover objetos horizontal o verticalmente.
- Editar: Cambiar posición, giro, seleccionar una cara del objeto...
- Crear: Primero hacer clic sobre la figura que queremos construir en el menú y luego clic en el lugar del mundo en el que queremos que aparezca.
- Terreno: Modificar las propiedades del terreno (altura, suavizado, escarpado...), dividir en parcelas y cambiar el nombre de las mismas.

#### **4.3.1.4 Programación para Opensim**

Una vez conocido el mecanismo para crear objetos, podemos hacer que estos ejecuten ciertas acciones e interactúen con el entorno, para ello usaremos los **scripts**. Estos scripts se programan en el lenguaje LSL (Linden Scripting Language) y están alojados en el inventario de los objetos. Un aspecto importante es que el código de los scripts se desarrolla en el visor, pero se ejecuta en el servidor.

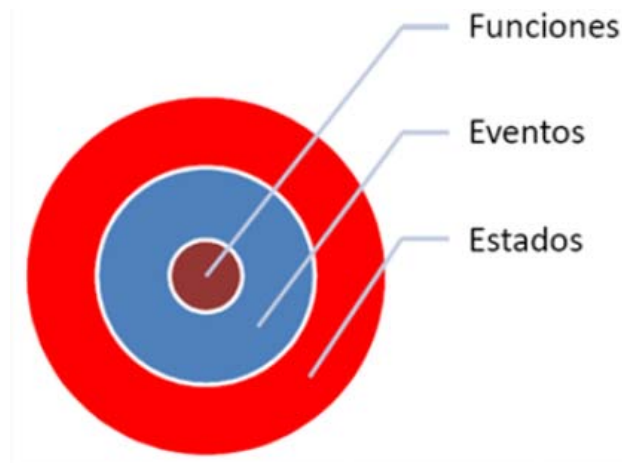
La programación del mundo virtual es abierta y libre. El código permite la creación y manipulación de programas o scripts, para poder modificar diferentes aspectos del entorno, desde hacer que un objeto se mueva, escuche, hable, actúe como un vehículo, cambie de color, tamaño o forma, etc.

## **Linden Scripting Language (LSL)**

Linden Scripting Language es un lenguaje basado en Java y C orientado a eventos y estados, permite dotar a los objetos de un cierto comportamiento y puedan interactuar con el avatar o con otros objetos.

Los scripts están compuestos de uno o varios estados, con los que podemos controlar los diferentes eventos (tocar, escuchar, temporizador...). Dentro de cada evento utilizamos las funciones (enviar un mensaje, detectar el avatar que ha tocado el objeto...), las cuales pueden ser tanto las que nos ofrece el lenguaje de programación como las que nosotros mismos podamos definir.

En la Figura 4.3-8 se muestra un esquema de la estructura de programación en LSL:



**Figura 4.3-8: Estructura del lenguaje LSL**

### **LSL Eventos**

LSL es un lenguaje manejado por eventos. Literalmente, los scripts se dividen en bloques de código que se activan cuando ocurre un determinado evento.

Si en un script se codifica un evento, y un evento de este tipo ocurre, el simulador añadirá este evento a la cola de eventos del script, siendo ejecutados en el orden en que han sido encolados en una cola tipo FIFO (First In First Out).

## LSL Estados

Los estados determinan cómo se comportará el script en cada momento ante la llegada de eventos o entradas del exterior. Cada estado definido en el script debe contener por lo menos un evento.

En LSL, los scripts están en espera hasta que reciben algún evento, o detectan algún cambio del exterior. En todo momento el script está en un estado determinado y reaccionará a los eventos o entradas de acuerdo con un esquema definido por el programador.

Todos los scripts deben tener definido el estado “*default*” al comienzo, que es el primer estado que se ejecuta cuando el script comienza. Cuando un script es compilado, reseteado o cargado, entra al estado “*default*”, y después de su declaración en el código pueden definirse otros estados adicionales.

## LSL Funciones

Mediante el uso de funciones, los mismos bloques de código se pueden usar una y otra vez, simplemente referenciándolo. Existen dos tipos de funciones:

1. **Funciones de Linden Labs:** LSL contiene más de 300 funciones incorporadas que permiten a los scripts y objetos interactuar entre ellos o con el medio exterior. Todas las funciones definidas en la librería Linden empiezan con "ll", que significan "Library Linden". El listado completo se encuentra en el wiki de secondlife: [http://wiki.secondlife.com/wiki/LSL\\_Portal](http://wiki.secondlife.com/wiki/LSL_Portal)
2. **Funciones OS:** Las funciones OS son algunas funciones específicas de OpenSim que pueden utilizar los scripts. No funcionan en Second Life y es necesario configurar el servidor (el fichero de configuración *OpenSim.ini*) para poder usar este tipo de funciones. La lista completa de estas funciones se encuentra en: [http://opensimulator.org/wiki/OSSL\\_Implemented](http://opensimulator.org/wiki/OSSL_Implemented).
3. **Funciones definidas por el usuario:** el usuario puede definir funciones (con valores de retorno, si se quiere), siempre que el nombre de la función del usuario no

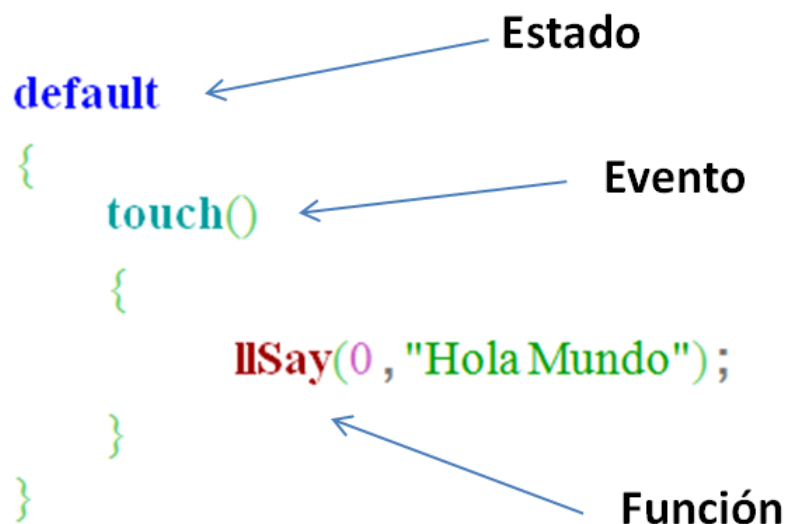
esté en conflicto con una palabra reservada por Linden, una constante reservada por Linden, o una función incorporada de la librería Linden.

## **Programación de Scripts**

Ya se conocen las herramientas a utilizar en la programación de scripts, ahora se comentarán algunos aspectos básicos.

Los scripts se asocian a objetos creados en el mundo virtual. Pueden adjuntarse varios scripts a un mismo objeto, lo que permite tener un conjunto de pequeños scripts con funciones simples, que combinados permiten formar nuevos comportamientos más complejos.

En la Figura 4.3-9 se muestra un código de ejemplo en LSL. Dicho código se ejecutaría cuando un avatar tocara el objeto que contiene este script, mostrando en el área de chat del visor el mensaje: "Hola Mundo".



```
default
{
    touch()
    {
        llSay(0, "Hola Mundo");
    }
}
```

El diagrama muestra un código LSL con tres anotaciones y flechas azules:

- Estado**: apunta a la palabra clave `default`.
- Evento**: apunta a la función `touch()`.
- Función**: apunta a la línea de código `llSay(0, "Hola Mundo");`.

**Figura 4.3-9: Ejemplo de un script en LSL**

En la Figura 4.3-10 se muestra la ventana de programación y el objeto que contiene la script de la Figura 4.3-9, dentro del mundo virtual.



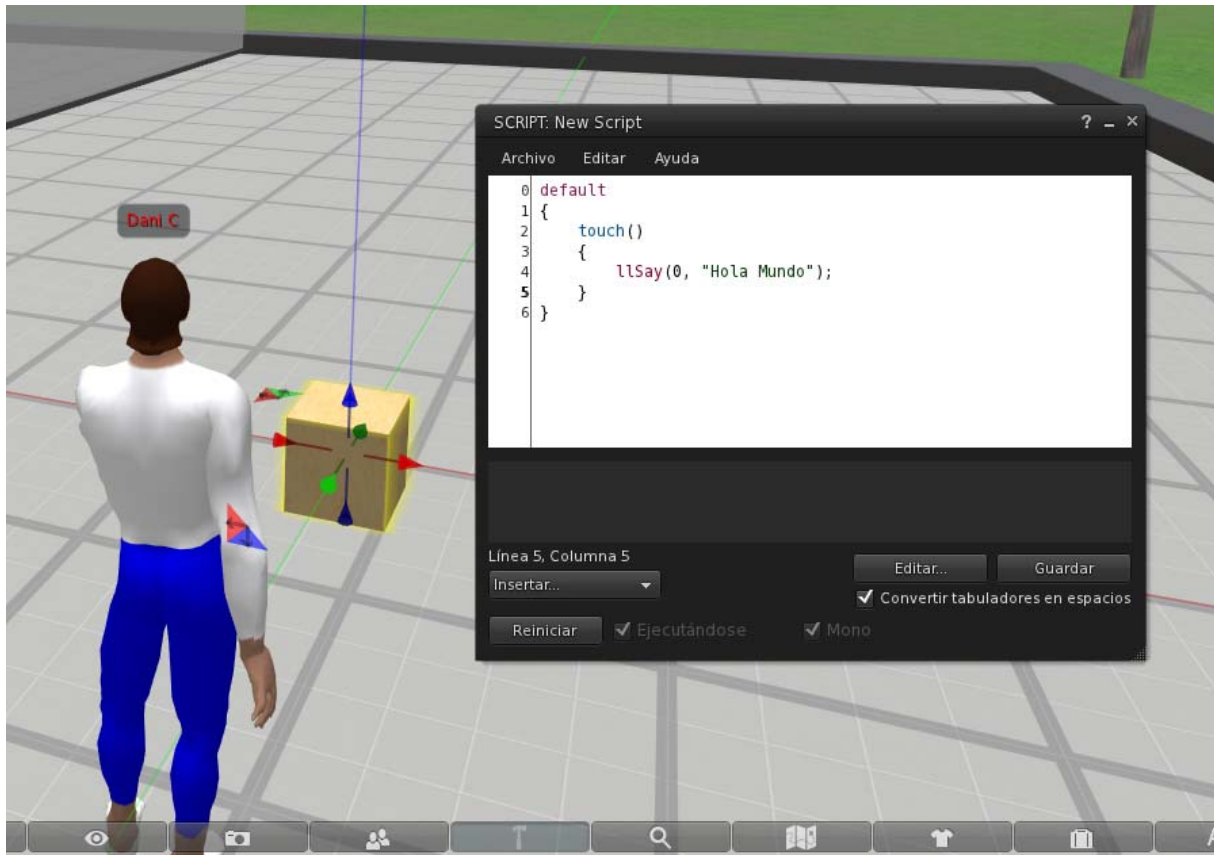


Figura 4.3-10: Objeto en mundo virtual con su script

## Comunicación entre scripts

Una de las mayores problemáticas que hay a la hora de crear el mundo virtual es la comunicación entre los objetos, mejor dicho, entre scripts ejecutándose en diferentes objetos. Por ejemplo, si un avatar toca el pomo de una puerta para que esta se abra, el pomo es quien tiene que comunicarle a la puerta que el usuario quiere abrirla, por lo que es necesaria la existencia de comunicación entre objetos.

Existen dos maneras de comunicar scripts entre objetos diferentes, mediante la comunicación por canales con función llSay(), o con comunicación directa entre objetos enlazados utilizando la función llMessageLinked().

1. **Comunicación mediante llSay():** Esta función envía un mensaje por el chat. El chat tiene muchos canales, de los cuales el único canal público es el 0 (que se utiliza para la comunicación entre avatares), por lo tanto, los mensajes que se



envíen por el resto de canales son invisibles para los avatares. Un ejemplo de esta función para enviar un mensaje por el canal 40 sería: **llSay( 40, "Esto es un mensaje por el canal 40" );**

Para que desde un script se puedan tratar los mensajes de chat enviados con la función **llSay()**, hay que abrir la escucha con la función **llListen(canal\_de\_escucha,"","")**. Esta función especifica el canal de escucha, y al recibir un mensaje se ejecuta el evento **listen**. Dentro de una misma script puede haber varios canales de escucha, uno por cada función **llListen** que se declare. Uno de los inconvenientes de esta función es que, si se utiliza en masa, consume muchos recursos. A continuación se presenta un ejemplo (Figura 4.3-11), que abre una escucha por el canal 40 y en el momento de escuchar algo por dicho canal se ejecuta el evento *listen*:

```
// Canal por el que escucharemos, lo establecemos como constante por si se quiere modificar
integer CANAL_COMUNICACION = 40;

default // Estado por defecto
{
    state_entry() // Evento inicial que se ejecuta al compilar el script
    {
        // Abrimos la escucha por dicho canal, cuando reciba algún mensaje, se lanza el evento 'listen'
        llListen (CANAL_COMUNICACION, "", "", "");
    }
    // Evento que se lanza al escuchar un mensaje por el canal de comunicacion especificado
    listen(integer channel, string name, key id, string message)
    {
        // En este evento generamos las respuestas a los mensajes recibidos por el canal de escucha abierto
        llSay(0, "El usuario " + name + " me ha dicho " + message + " por el canal " + channel);
    }
}
```

**Figura 4.3-11: Código de ejemplo de la función llSay**

Un aspecto a considerar es que esta función tiene un límite de distancia para que los objetos escuchen los mensajes. Este límite por defecto es 20 metros, aunque en el fichero **OpenSim.ini** se puede modificar.

2. **Comunicación mediante llMessageLinked():** Este método es más recomendable en objetos unidos/dependientes, ya que no utiliza la comunicación mediante canales, sino que al estar unidos tienen comunicación directa. Llamando a la función **llMessageLinked()** se lanzará un evento **link\_message** en los demás objetos del grupo enlazado. Esto consume muchos menos recursos.

A continuación se muestra un ejemplo de esta función (Figura 4.3-12). Lo que realiza esta función es, nada más arrancar (*state\_entry*) envía un mensaje a el resto de objetos enlazados, y mediante el evento *link\_message* escucha los mensajes que envíen los demás objetos enlazados:

```
default // Estado por defecto
{
    state_entry() // Evento inicial que se ejecuta al compilar el script
    {
        // Con esta función podemos enviar tres tipos de datos: un entero, un string y un tipo key.
        llMessageLinked(LINK_ALL_OTHERS, 40, "Script ejecutándose", llGetKey());
    }
    link_message(integer sender_num, integer num, string str, key id)
    {
        // Cuando recibimos un mensaje de otro objeto linkado se lanza este evento.
        llSay(0, "El objeto con índice " + sender_num + " y con UUID " + id +
            "ha enviado el número " + num + " y el mensaje " + str);
    }
}
```

**Figura 4.3-12: Código de ejemplo de la función llMessageLinked**

A la hora de crear la región del laboratorio de electrónica, uno de los problemas que se detectó era poder tener un sistema de comunicación entre objetos sencillo y sobre todo modular, ya que contamos con un número muy elevado de ellos. Un puesto de laboratorio tiene alrededor de las 300 primitivas.

A simple vista, lo más simple parece utilizar un único canal privado, pero esta opción ralentizaría mucho el sistema ya que todos los objetos escucharían todos los mensajes, y no es necesario que haya comunicación entre todos.

Toda la problemática surge por ser un laboratorio remoto y no virtual. En el mundo virtual se tienen varios puestos de laboratorio disponibles (todos iguales y con los mismo objetos), pero físicamente solo hay un equipamiento real que controlar remotamente (un multímetro, una fuente de alimentación...). Por ello, cada puesto de laboratorio del mundo virtual debe comunicarse con el servidor laboratorio y entre todos los objetos que lo forman, pero no debe interferir en los demás puestos.

La forma de comunicación (que se explica al detalle en el apartado *Comunicación entre Opensim y aplicaciones externas*), consiste en realizar una petición por cada

cambio de configuración que realice un usuario, teniendo en cuenta que algunos cambios repercuten en los demás equipos, pero otros no. La opción que se ha escogido es la de utilizar un canal por cada equipo y todo lo relacionado con él (cableado, conexiones...), reduciendo considerablemente el flujo de mensajes.

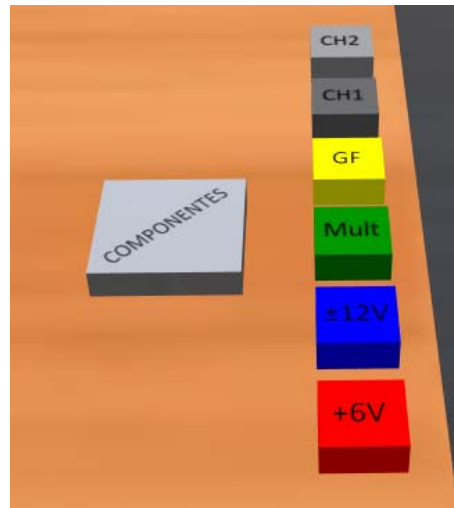
Pongamos un ejemplo para aclarar mejor esta problemática y la solución que se ha llevado a cabo. La mayor parte del código desarrollado en el laboratorio se basa en esta solución.

Si cambiamos la excitación de entrada de un circuito, por ejemplo el generador de señales, las medidas que se estuvieran tomando en ese momento deberían variar. Por lo tanto al cambiar la configuración del generador, tanto el osciloscopio como el multímetro deben enterarse de dicho cambio. Esto sería sencillo de implementar si por cada puesto de laboratorio del mundo virtual tuviéramos uno real, ya que la comunicación sería siempre directa y podríamos estar recogiendo la información de cada instrumento en tiempo real.

Al no ser así, aparte de tener un canal por cada equipo, los canales deben ser diferentes entre los mismos equipos de puestos de laboratorios diferentes para que no interfiera la configuración de un usuario con otro.

Para solucionar todo esto, se han desarrollado unos **registros**, siendo una de sus funciones encargarse de recoger todas las acciones que realiza el usuario de un puesto de laboratorio y comunicar dichas acciones entre los objetos que intervengan, o en los que repercuta.

Los registros que se han creado se muestran en la Figura 4.3-13:



**Figura 4.3-13: Registros de comunicaciones**

Como se aprecia en la Figura 4.3-13, se distinguen dos tipos de registros, los relacionados con los equipos y un registro de componentes. En el apartado anterior se habló de los objetos de una región, y se comentó que había objetos que se creaban dinámicamente según fueran a utilizarse. Pues bien, esa es otra de las funciones de estos registros: crean y mueven los cables de los equipos, y colocan y gestionan los componentes utilizados en la práctica.

La mayor complejidad de estos registros era poder separar la comunicación y control de cada equipo para no sobrecargar el sistema de mensajes. Al tener registros independientes la depuración se simplifica y se minimiza la carga del sistema, ya que cada uno de ellos atiende solo las acciones que repercuten en ellos.

Otra ventaja importante es que al tener registros en cada puesto solucionas el problema de la interferencia de unos puestos con otros, ya que en cada puesto los registros son idénticos, pero escuchan canales diferentes.

### **Comunicación entre Opensim y aplicaciones externas**

LSL tiene protocolos de comunicación externa http y ftp, así nuestro programa no se queda sólo dentro del mundo virtual donde se esté ejecutando, sino que puede comunicarse con una web, un servidor, una aplicación, etc.

Las opciones de comunicación externa de LSL están limitadas a tres opciones específicas. El wiki oficial de LSL proporciona información más detallada sobre cada opción:

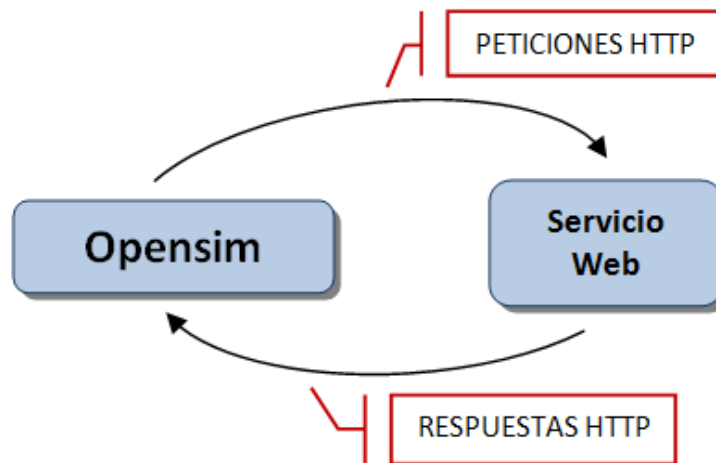
- **HTTP:** peticiones http que deben ser iniciados por script LSL.
- **Xml-RPC:** peticiones que deben ser iniciados por servicio externo.
- **Correo electrónico:** comunicación bidireccional, pero con temporizadores de espera forzada.

Esta comunicación se realiza mediante funciones propias de LSL, por lo tanto la comunicación sería desde objetos alojados en el mundo virtual con una aplicación externa.

La mejor opción es realizar la comunicación mediante HTTP, utilizando la función **IIHTTPRequest** y evento **http\_response**. Más adelante se explicará el funcionamiento y se detallará un ejemplo.

Aquí se encuentra una de las partes mas importantes y novedosas de todo el sistema software, ya que teníamos la problemática de cómo comunicar el servidor Opensim con el servidor del laboratorio encargado de la gestión de usuarios y sistema hardware, que está implementado con el lenguaje de programación LabView.

Gracias a la comunicación HTTP que proporciona LSL solucionamos la problemática entre las dos partes, implementando un **Servicio Web** en LabView (descrito en el apartado 4.3.2.1 *Servicio Web*). Este servicio web atiende las peticiones HTTP procedentes de las acciones de los avatares dentro del mundo virtual, realizadas mediante la función **IIHTTPRequest**. Una vez procesada dicha petición por el servidor laboratorio, el propio servicio web es quien responde. Esta respuesta se recoge mediante el evento **http\_response** para ser procesada en el mundo virtual. El diagrama de flujo de la comunicación se muestra en la Figura 4.3-14:



**Figura 4.3-14: Comunicación Opensim-Servicio Web**

Un ejemplo de un script con la función `llHTTPRequest` y el evento `http_response` se muestra en la Figura 4.3-15. Cuando un avatar toque el objeto que contiene esta script, se realiza una petición a una URL particular con la función `llHTTPRequest()`. La respuesta de la petición es manejada en el evento de respuesta, `http_response()`, el cual comprueba que la petición se haya realizada con éxito y los datos pedidos son enviados mediante formato HTML y convertidos en una cadena de texto almacenada en el parámetro `body`, el cual es mostrado en pantalla por el chat público, con la función `llSay()`.

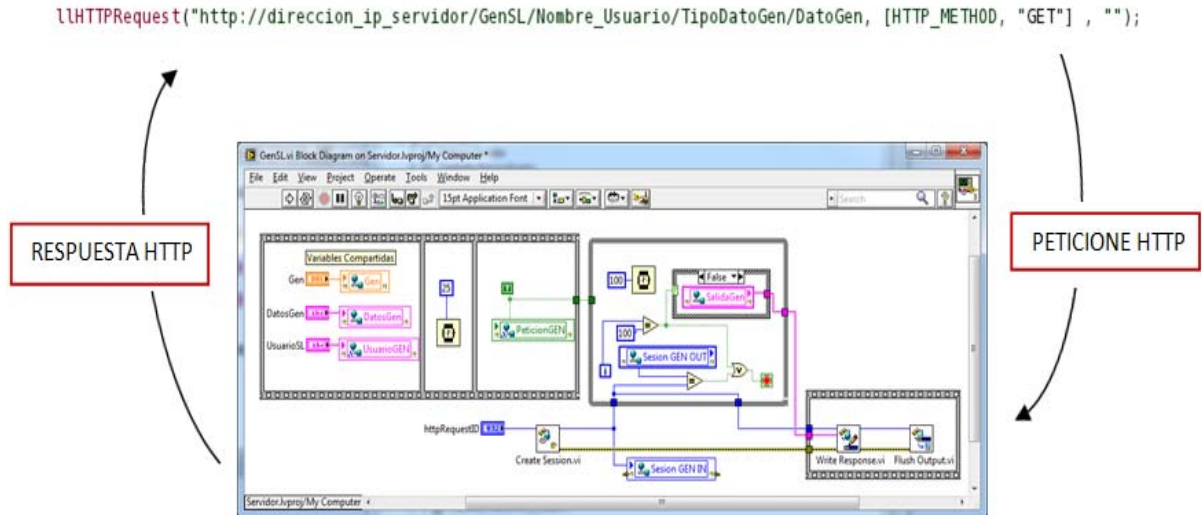
```

key http_request_id; // Identificador de peticion http
default
{
    // Evento que se ejecuta al tocar el objeto que contiene el script
    touch(integer x)
    {
        http_request_id = llHTTPRequest("URL_DESTINO", [HTTP_METHOD, "GET"] , "");
    }
    // Evento que se ejecuta recibir la respuesta de la peticion http
    http_response(key request_id, integer status, list metadata, string body)
    {
        if ( request_id == http_request_id )
        {
            // Si coincide con la petición realizada en el evento touch muestra
            // el cuerpo del mensaje recibido
            llSay(0, "Body recibido: "+body);
        }
    }
}

```

**Figura 4.3-15: Ejemplo de script con función `llHTTPRequest`**

En concreto, en la región del laboratorio de electrónica, la comunicación con el servidor laboratorio mediante el servicio web se realiza como indica la Figura 4.3-16, que es una petición al servicio web del generador de funciones:



**Figura 4.3-16: Ejemplo comunicación mediante http\_request**

Como se comentó anteriormente, la petición realizada por el generador de funciones repercute en los demás equipos, por lo que el objeto que recibe la respuesta, es el encargado de comunicársela, mediante los registros de comunicaciones anteriormente explicados, al osciloscopio y multímetro para que muestren los nuevos valores.

#### 4.3.1.5 *Requerimientos técnicos para el uso de Opensim*

En función del sistema operativo de nuestro servidor, hay que tener en cuenta los siguientes aspectos:

- Opensim puede ejecutarse en sistemas operativos de 32 y 64 bits.
- Es necesario disponer de una base de datos MySQL, en la que se guardará toda la información del simulador. Por defecto OpenSim se ejecuta en una SQLite pero usando MySQL obtenemos mayor rendimiento, funcionalidad y versatilidad.
- Para ejecutar en Windows es necesario:

- NET Framework 3.5. En versiones anteriores y en NET Framework 4.0 no se asegura su funcionamiento.
- Si se ejecuta en Windows XP asegurarse de que esté actualizado, al menos el Service Pack 2 (SP2).
- Dependiendo de la instalación, puede que sea necesario ejecutar el programa como administrador (clic derecho -> "Ejecutar como administrador"). Se mostrará una ventana pidiendo permiso, seleccionamos "Permitir".
- Para ejecutar en Linux y Mac OSX es necesario:
  - Usar Mono 2.4.3, 2.6.x, 2.10.2 o posterior.

#### **4.3.1.6    *Mundo Virtual del Laboratorio de Electrónica***

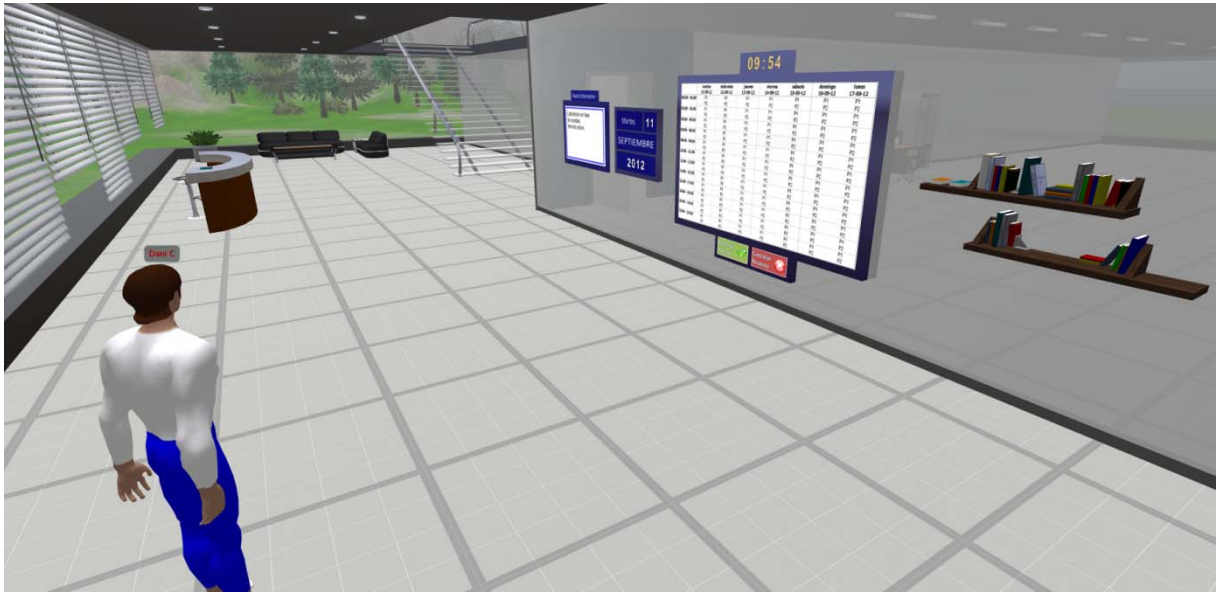
Una vez definidas las funcionalidades del servidor de mundos virtuales Opensim, y como se crea y se dota de funcionalidad el mundo virtual, se muestran una serie de imágenes del laboratorio de electrónica. En el apartado 5 *MANUAL DE USUARIO I: VISOR 3D*, se muestra al detalle el laboratorio creado, tanto los instrumentos como su cableado, las placas de prueba y sus componentes, etc.

En la Figura 4.3-17 se muestra el edificio creado para el laboratorio de electrónica, y en la siguiente figura (Figura 4.3-18) se observa su entrada:



**Figura 4.3-17: Edificio Laboratorio Electrónica**





**Figura 4.3-18: Entrada del laboratorio de electrónica**

En la Figura 4.3-19 se muestra como es un puesto de laboratorio para realizar las prácticas:



**Figura 4.3-19: Puesto de laboratorio**

### 4.3.2 Aplicación servidor

Como se mostró en la Figura 4.1-1, esta parte del sistema software del servidor se dividía en dos partes (Figura 4.3-20):



**Figura 4.3-20: Partes del servidor de laboratorio**

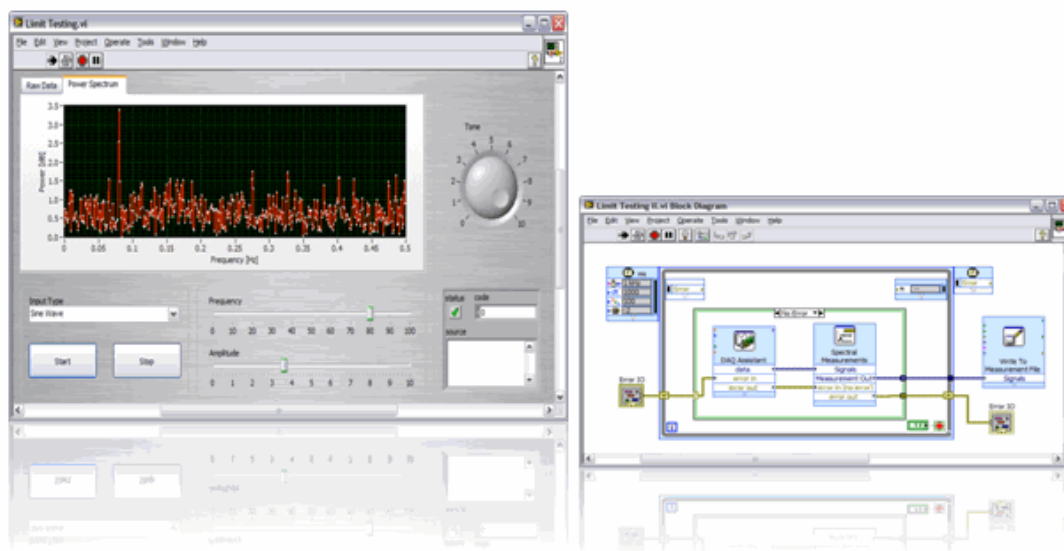
Antes de abordar el diseño realizado se define brevemente el programa utilizado para llevar a cabo estas dos partes.

**LabVIEW** (Laboratory Virtual Instrument Engineering Workbench) es un entorno de programación destinado al desarrollo de aplicaciones, empleando la programación gráfica o lenguaje G para crear programas basados en diagramas de bloques, diseñado para automatizar la adquisición, análisis y presentación de datos. Este programa fue creado por la empresa National Instruments.

Los programas en LabVIEW se conocen como instrumentos virtuales o VIs. Los VIs tienen 2 ventanas principales (Figura 4.3-21):

1. **Panel Frontal:** es la interfaz con el usuario, la utilizamos para interactuar con el usuario cuando el programa se está ejecutando. Los usuarios podrán observar los datos del programa actualizados en tiempo real. En esta interfaz se definen los controles (entradas) e indicadores (salidas).

2. **Diagrama de Bloques:** es el programa propiamente dicho, donde se define su funcionalidad. Aquí se colocan e interconectan íconos que realizan una determinada función.



**Figura 4.3-21: Panel frontal y diagrama de bloques de un programa de LabView**

#### 4.3.2.1 *Servicio Web*

Un servicio web (o Web Service) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas con lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos a través de Internet.

Por lo tanto, los servicios Web son programas que pueden ser llamados de forma remota en una red usando HTTP y permiten a los desarrolladores utilizar clientes web personalizados para mostrar información y controlar aplicaciones de LabVIEW. En nuestro caso lo que utiliza como cliente web es el visor 3D del mundo virtual, pero no es el visor quien realiza las peticiones, si no el servidor Opensim.

Cabe aclarar aquí que una vez creado el servicio web está siempre activo, independientemente de si el servidor laboratorio lo está. El servicio web puede atender a peticiones procedentes del servidor Opensim referentes a las acciones del usuario, pero si el servidor laboratorio no está activo, dichas peticiones no se procesarán. Si se da este caso, se avisa al usuario de que no está activo.

## ARQUITECTURA

Actualmente dos de las arquitecturas de servicios Web más extendidas son REST y SOAP. La arquitectura escogida para llevar a cabo el servicio web es **REST**, por dos motivos:

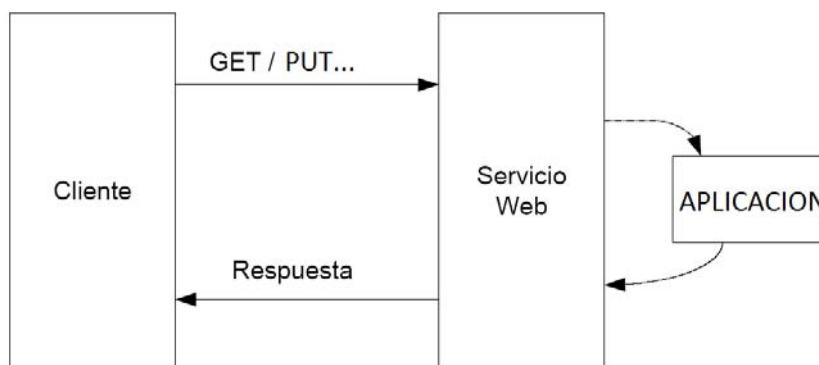
- El servidor del mundo virtual Opensim utiliza HTTP.
- LabView puede implementar servicios web mediante RESTful, que se basa en el protocolo HTTP.

A continuación describiremos brevemente la arquitectura REST para servicios web.

Servicios Web REST: es una arquitectura enfocada a acceder a los recursos de una manera sencilla y sin estado. Sin estado implica que no se almacena (ni en el cliente, ni el servidor) el estado de la comunicación.

RESTfull HTTP utiliza los métodos fundamentales de HTTP: GET, PUT, POST y DELETE. Estos métodos permiten realizar las operaciones de recuperar, crear, actualizar y borrar recursos en el servidor. Estos recursos son ofrecidos en Internet y se identifican mediante un identificador único de recurso (URI) o una URL para referenciar al recurso.

En la Figura 4.3-22 se muestra la secuencia de una comunicación con un servicio web mediante REST:



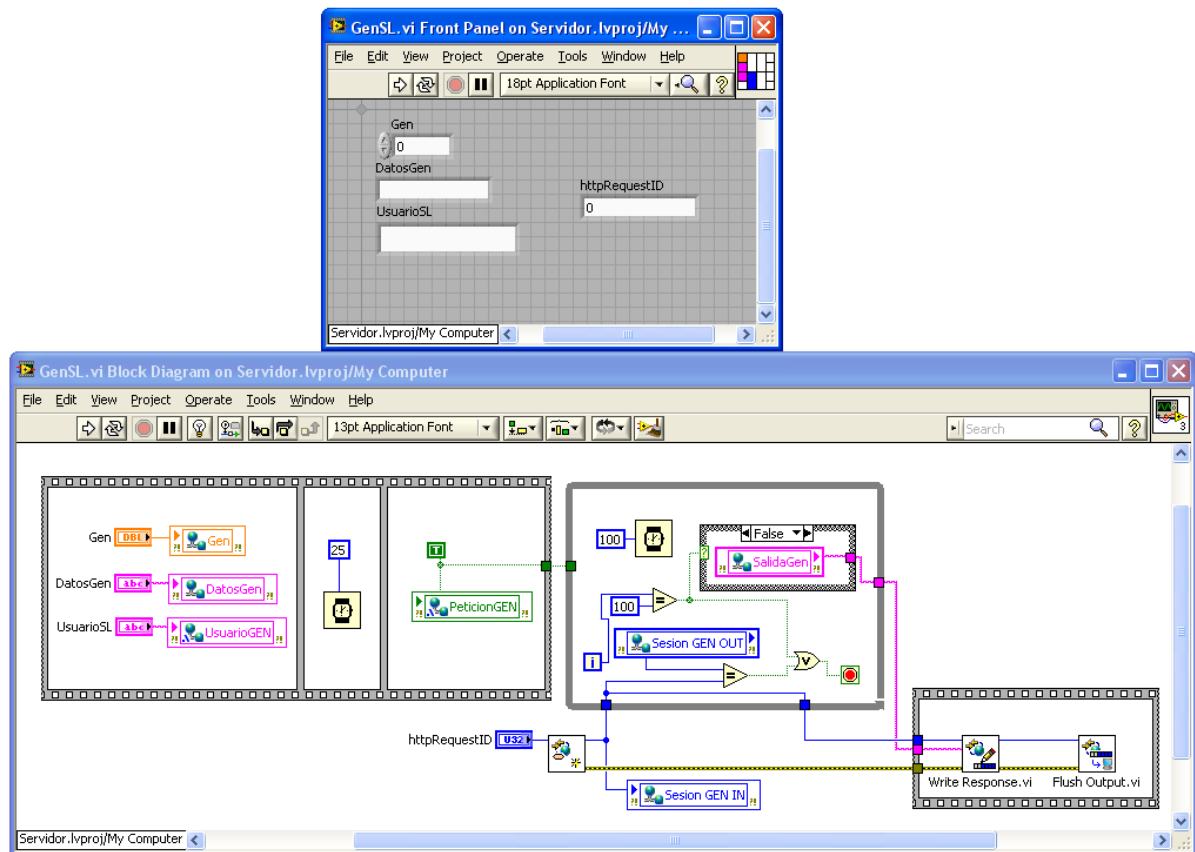
**Figura 4.3-22: Llamada a un servicio web**

## CREACIÓN DEL SERVICIO WEB EN LABVIEW

A continuación se explica paso a paso (a modo de manual) las acciones a realizar para crear un servicio web en labview. En concreto, se muestra la creación del servicio web a partir del VI que atiende las peticiones relacionadas con el generador de señales. Los demás Vis pertenecientes al servicio web creado han seguido los mismos pasos.

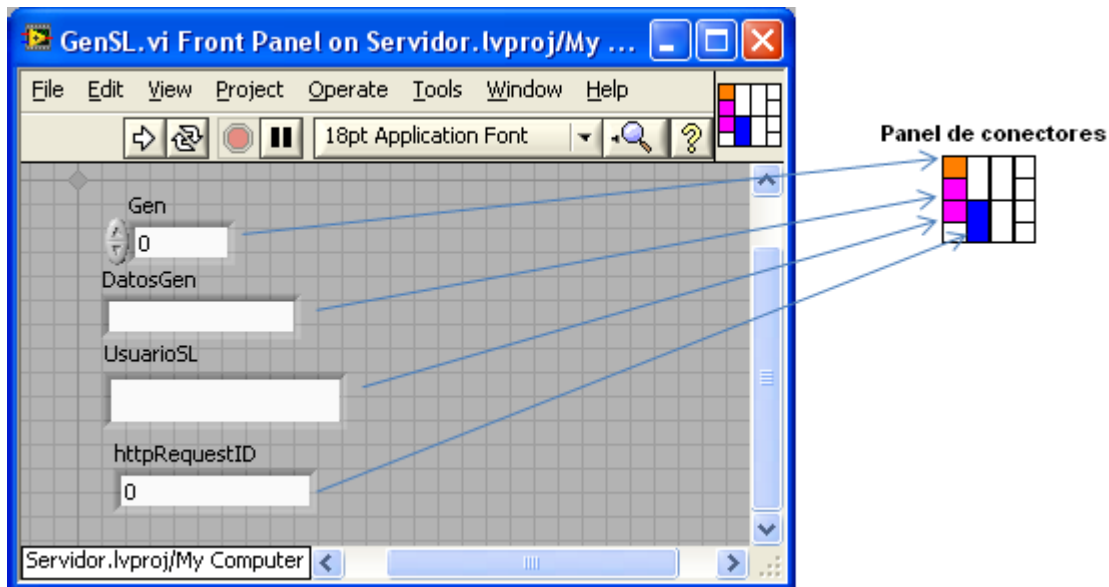
### Creación de un VI para implementar un servicio web

1. Crear un VI y asignarle el nombre **GenSL.vi**. Como ejemplo para crear el servicio web se ha cogido el VI del generador de señales, Figura 4.3-23:



**Figura 4.3-23: Figura 1 manual Servicio Web**

2. Asignación de los controles e indicadores del VI en el panel de conectores: una vez generado el código del VI, se debe asignar al panel de conectores los controles, que serán los encargados de recoger los datos del cliente web, y los indicadores, que se encargan de devolver la respuesta generada. En la Figura 4.3-24 se muestra la asignación realizada:



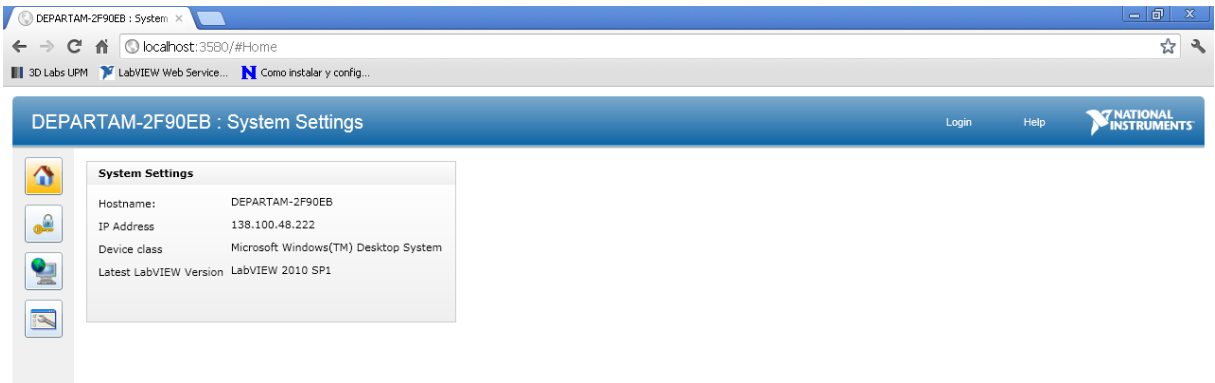
**Figura 4.3-24: Figura 2 manual Servicio Web**

3. Guardar el VI generado.



### **Configurar la aplicación del servidor Web de LabVIEW**

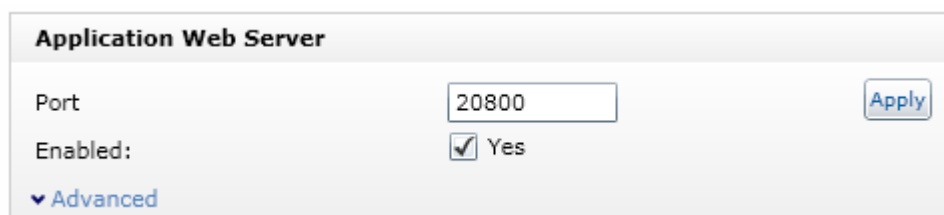
1. Abrir un navegador Web estándar.
2. Para visualizar Web Based-NI Supervisión y configuración (es el Servidor de Aplicaciones Web que utiliza labview), en el navegador escribir:
  - a. Para configurar el servidor de aplicaciones Web en el sistema local, introducir la siguiente URL: `http://localhost:3580`
  - b. Para configurar el servidor de aplicaciones Web en un sistema remoto, escribir la dirección URL siguiente: `http:// [dirección IP del dispositivo remoto]:3580`, donde [dirección IP del dispositivo remoto] es la dirección IP real del dispositivo remoto.
3. Si es necesario, instale el navegador de Microsoft Silverlight plug-in. LabVIEW requiere que el navegador de Microsoft Silverlight plug-in esté instalado para activar y configurar el servidor de aplicaciones Web.
  - a. Consultar el sitio Web de Microsoft en [www.microsoft.com](http://www.microsoft.com) para descargar el Microsoft Silverlight plug-in y seguir las instrucciones en pantalla para instalar el Microsoft Silverlight plug-in.

- b. Después de terminar la instalación, actualizar el explorador o vuelva a introducir la URL para visualizar Web Based-NI Supervisión y configuración.
- c. La pantalla que debe salir es la mostrada en la Figura 4.3-25:



**Figura 4.3-25: Figura 3 manual Servicio Web**

4. En la página mostrada hacer clic en el botón  en la esquina superior derecha.
5. En el cuadro de diálogo, introducir un nombre de usuario y contraseña para acceder. Si no tiene nombre de usuario o contraseña escriba el nombre de usuario admin y la contraseña en blanco para acceder con las credenciales predeterminadas.
6. Después de iniciar sesión, hacer clic en el botón  (**configuration Web Server**), en el panel izquierdo.
7. En **Application web server**, habilitar el servidor de aplicaciones Web.
8. En el cuadro de texto **Port**, introducir el puerto de red en el que se ejecutará el servidor de aplicaciones Web. Debe quedar como en la Figura 4.3-26:



**Figura 4.3-26: Figura 4 manual Servicio Web**

9. (Opcional) Habilitar la seguridad SSL para servicios web para cifrar la comunicación en el servidor de aplicaciones Web.

### Crear un proyecto de servicio Web y construir Especificaciones

1. Crear un Proyecto de LabVIEW para organizar y construir una aplicación de servicio Web.
2. Añadir el VI al proyecto que se desea implementar como un servicio Web, en este caso, **GenSL.vi**. Debe añadir cualquier VI que contenga controles o indicadores que intercambian datos a través de la red con clientes Web. LabVIEW incluye automáticamente cualquier subVI. En la Figura 4.3-27 se muestra el proyecto con los Vis donde se ha creado el servicio web:

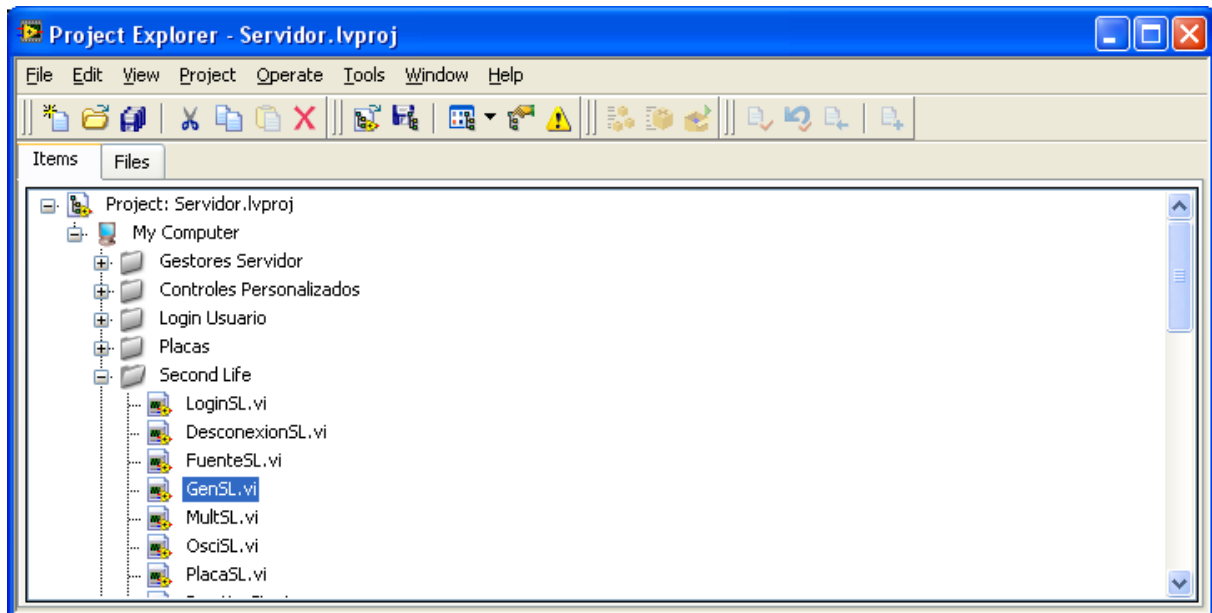
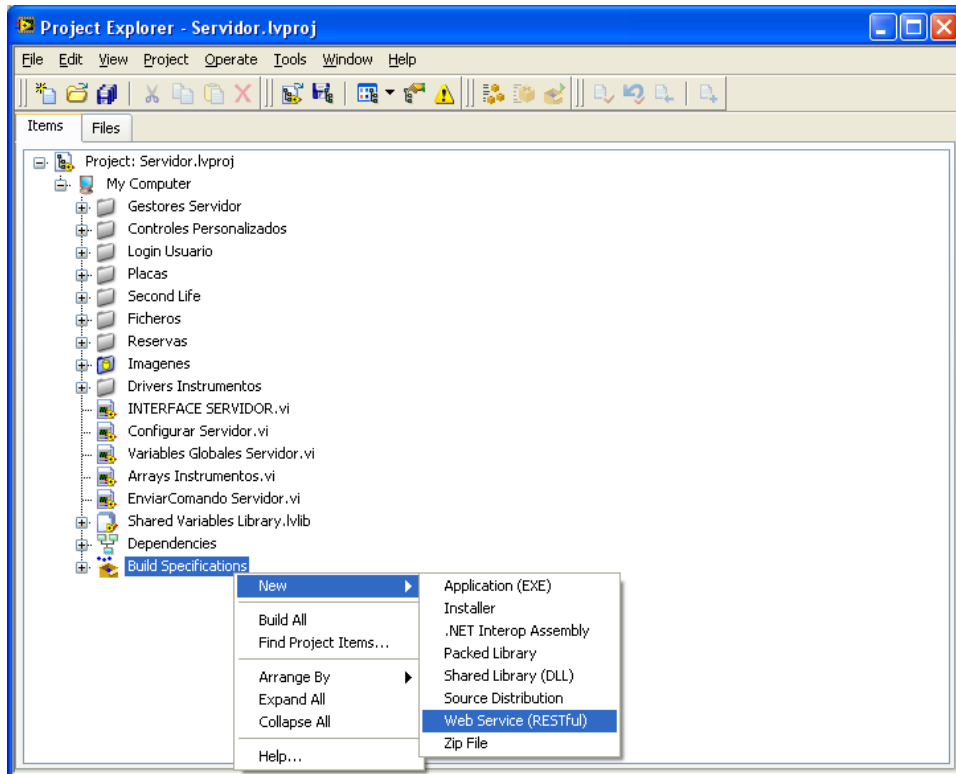


Figura 4.3-27: Figura 5 manual Servicio Web

3. (Opcional) Agregar las páginas HTML o imágenes para el proyecto que se desea implementar en el marco de la aplicación del servicio Web.
4. (Opcional) Agregar cualquier VI auxiliar ajeno al servicio web, pero que sea necesario para la ejecución del proyecto.
5. Guardar el proyecto. Utilizar el nombre que desee, en este caso **Servidor.lvproj**.
6. Hacer clic con el botón derecho del ratón en **Build Specifications** y seleccione **New »Servicio Web (RESTfull)** en el menú contextual. Aparecerá la ventana mostrada en la Figura 4.3-28:





**Figura 4.3-28: Figura 6 manual Servicio Web**

7. En la ventana que aparece (Figura 4.3-29), en el apartado **Information**, dentro de Web Service Properties:
  - a. Especificar un nombre en **Build specification name**. Este valor sólo aparece en el explorador del proyecto y debe ser único.
  - b. Especificar un nombre en **Service name**. Este será el nombre del servicio web que es parte de la URL que los clientes Web utilizan para intercambiar datos con el servicio Web.

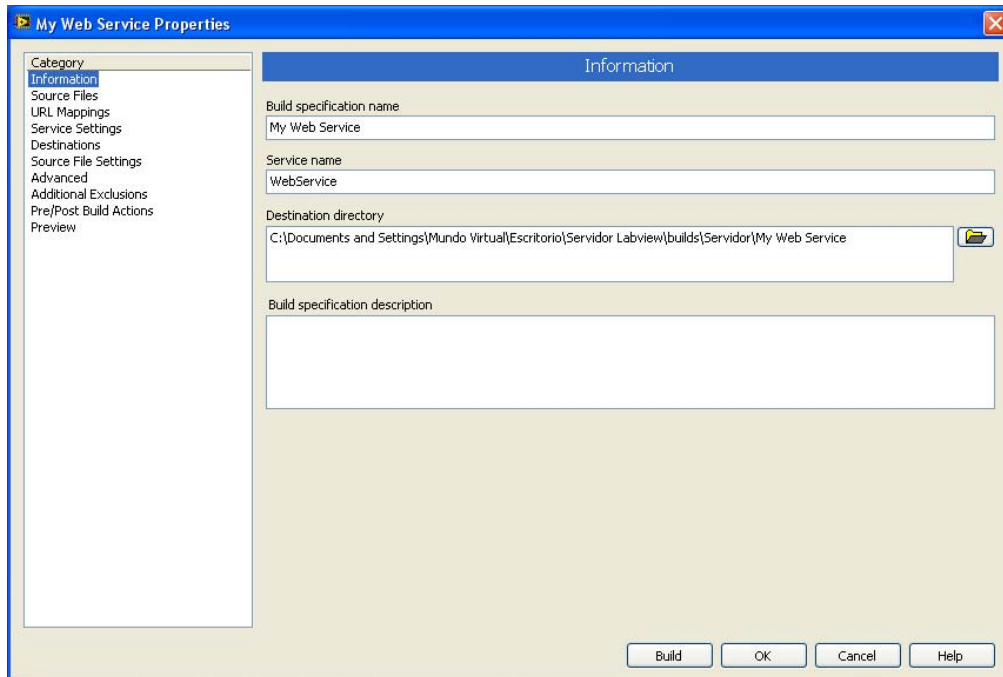



Figura 4.3-29: Figura 7 manual Servicio Web

8. En el apartado **Source Files** (Figura 4.3-30), realizar los pasos siguientes para incluir los VIs que se desea implementar dentro del servicio Web.
  - a. Utilizar la flecha  para mover los VIs incluidos en el proyecto, desde **Project Files** a **Service Vis**, para que sean ejecutados en el servicio web.

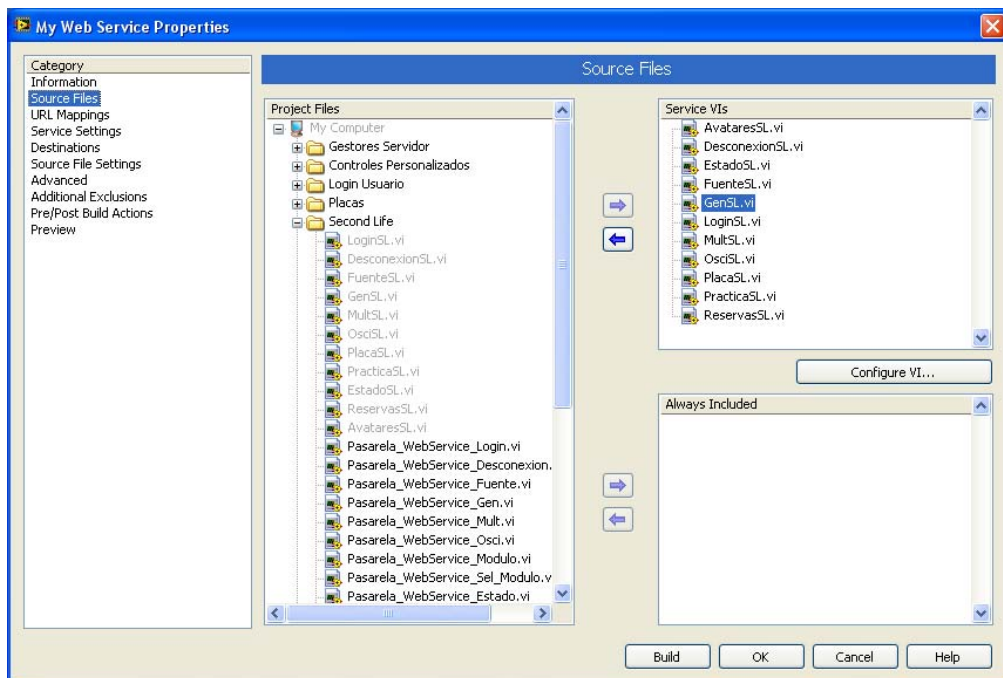
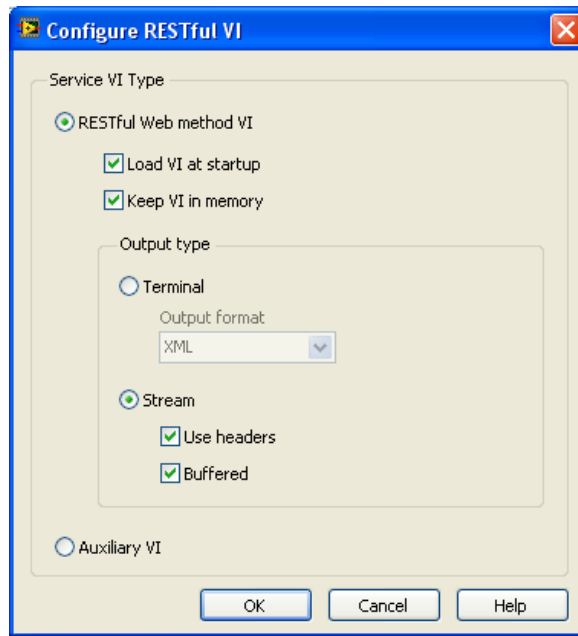



Figura 4.3-30: Figura 8 manual Servicio Web

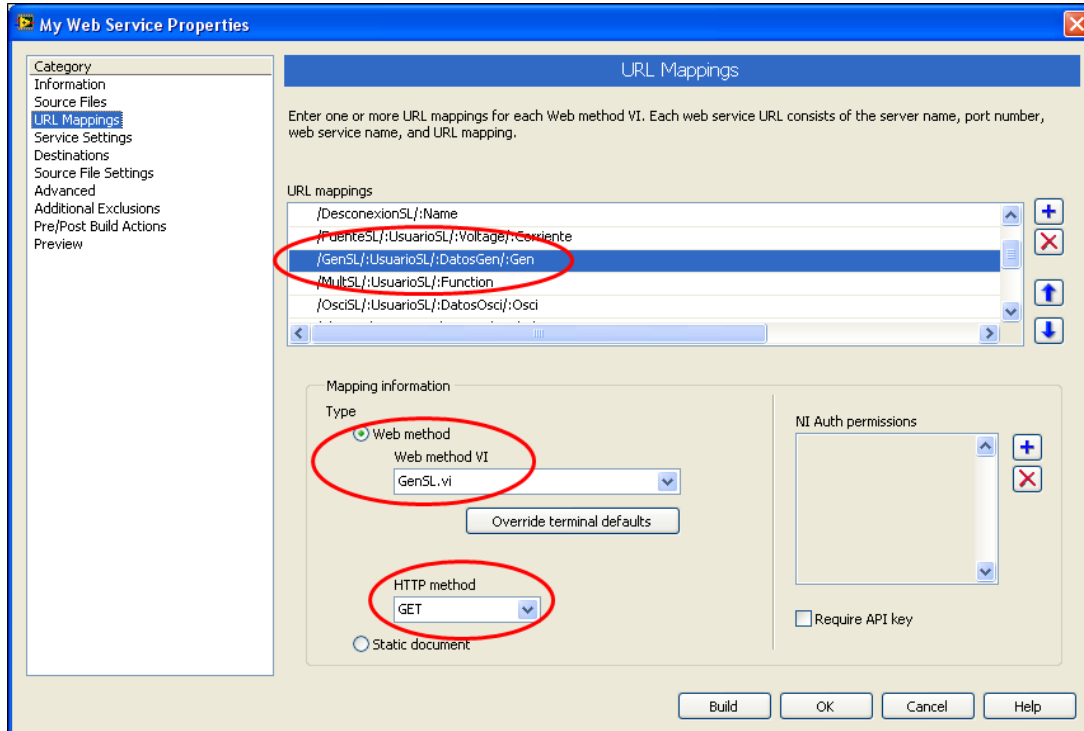
- b. Pulsando en el botón **Configure VI...**, aparece un nuevo menú que debe quedar configurado como en la Figura 4.3-31. En el cuadro de diálogo, puede configurar el VI como un método Web VI o un auxiliar VI, así como al formato del envío de las respuestas al cliente Web.



**Figura 4.3-31: Figura 9 manual Servicio Web**

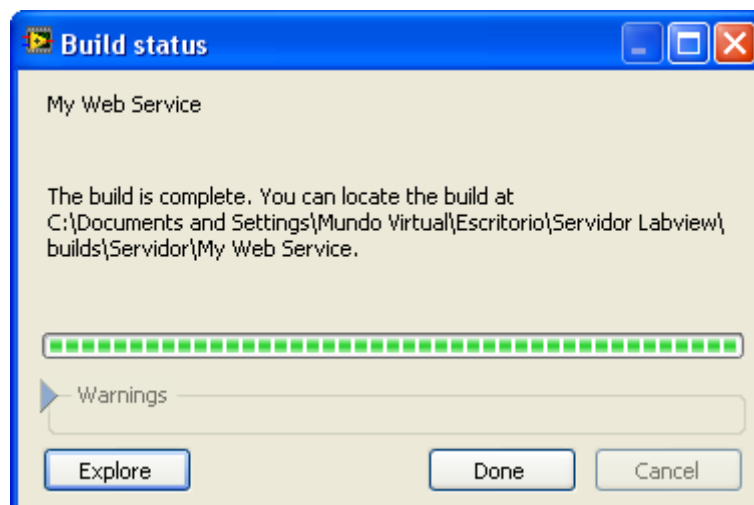
9. En **URL Mappings** aparecen las URL, creadas por defecto por Labview, de los VIs que se incluyeron en el paso 8.a. Los clientes web utilizan estas URLs para encaminar las peticiones a unos VIs u otros, y así asignar valores a los controles que contiene cada VI.
- a. Cuando se agrega un nuevo VI al **Service VIs** en el apartado **Source Files**, LabVIEW crea una asignación de dirección URL por defecto que incluye los controles del panel de conectores del VI. Para **GenSL.vi**, LabVIEW genera la asignación de dirección URL por defecto: **/GenSL:UsuarioSL/:DatosGen/:Gen**. Si una asignación de dirección URL por defecto no aparece, o si desea definir asignaciones adicionales de URL, con el botón  se pueden añadir o modificar.
- b. Seleccionar un método VI web, **GenSL.vi** en este caso, para asignar el tipo de acceso.
- c. En **Mapping Information -> Type**, se selecciona **Web Method** y se asigna el método al VI **GenSL.vi**.

- d. En **HTTP method** mediante el menú desplegable seleccionar el valor predeterminado GET.
- e. Debe quedar como aparece en la Figura 4.3-32:



**Figura 4.3-32: Figura 10 manual Servicio Web**

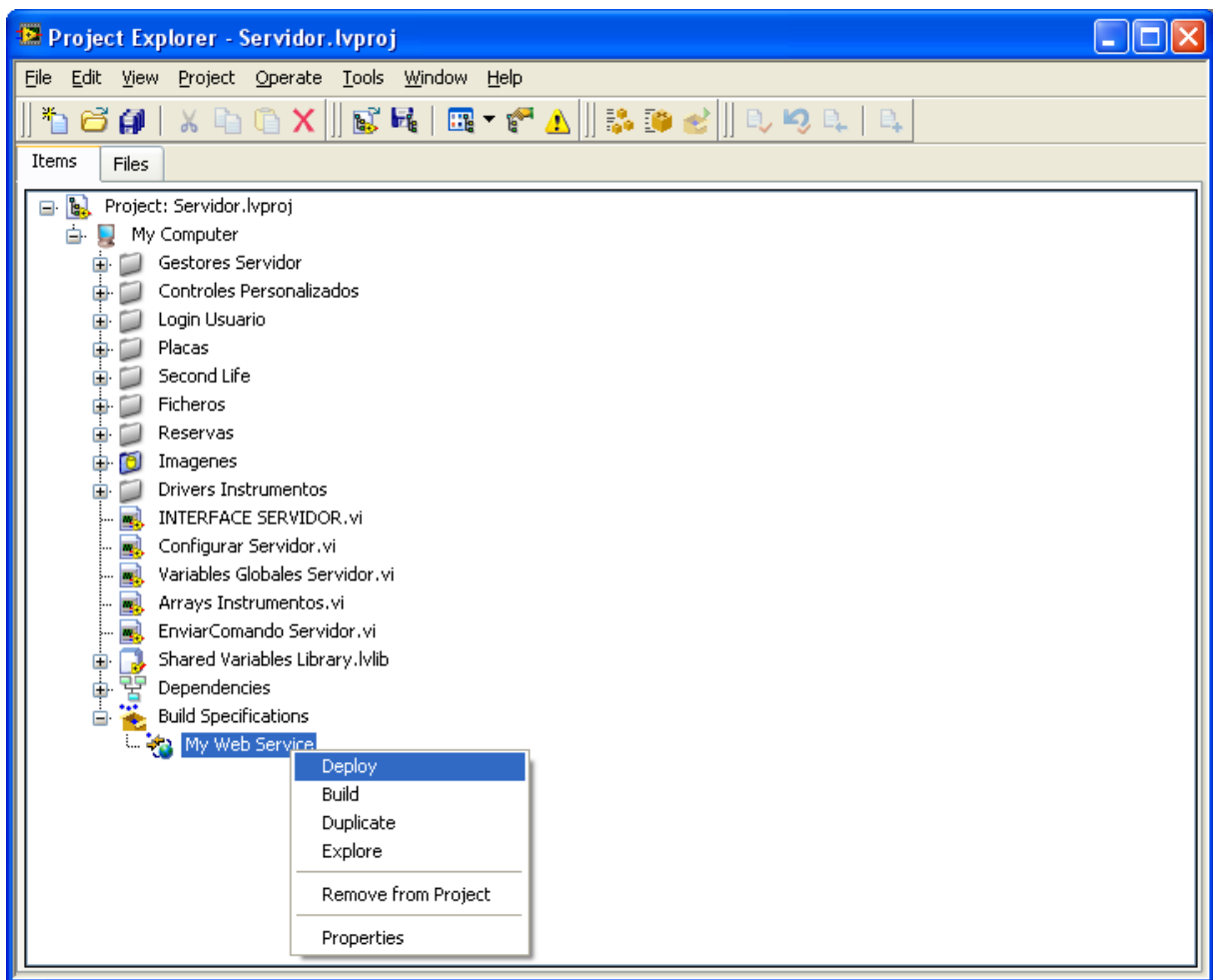
- 10. Hacer clic en el botón **Build** para crear el servicio Web. También está la opción de crear el servicio web desde la ventana de explorador del proyecto.
- 11. En la ventana que aparece (Figura 4.3-33), una vez terminado el proceso, hacer clic en **DONE** para finalizar la construcción del servicio Web.



**Figura 4.3-33: Figura 11 manual Servicio Web**

## Implementar un servicio Web en el servidor Web

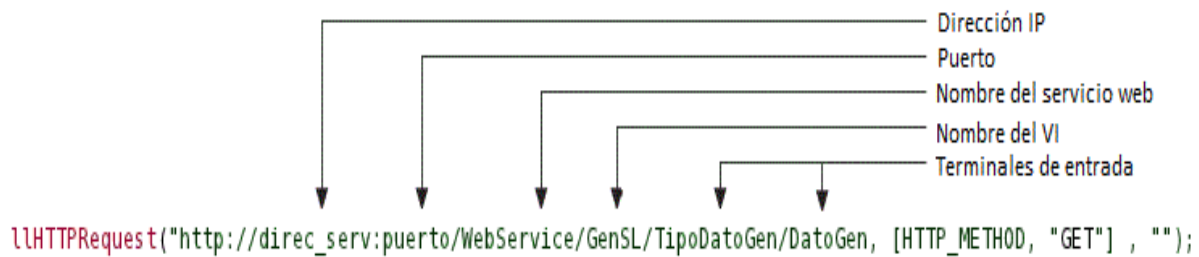
Después de crear el servicio Web, en el explorador del proyecto y dentro de **Build Specification**, aparece el servicio web creado (Figura 4.3-34). Para implementar y lanzar el servicio web, haga clic con el botón derecho del ratón sobre el nombre y seleccione **Deploy** en el menú contextual. Después de implementar el servicio web, los clientes Web pueden intercambiar datos con los VIs incluidos.



**Figura 4.3-34: Figura 12 manual Servicio Web**

## ACCESO AL SERVICIO WEB

El usuario genera una petición desde el cliente a una URL específica del servicio web creado. La Figura 4.3-35 muestra una URL de ejemplo, que el servidor Opensim envía para acceder a un VI declarado en el servicio web de Labview:



**Figura 4.3-35: Ejemplo de URL**

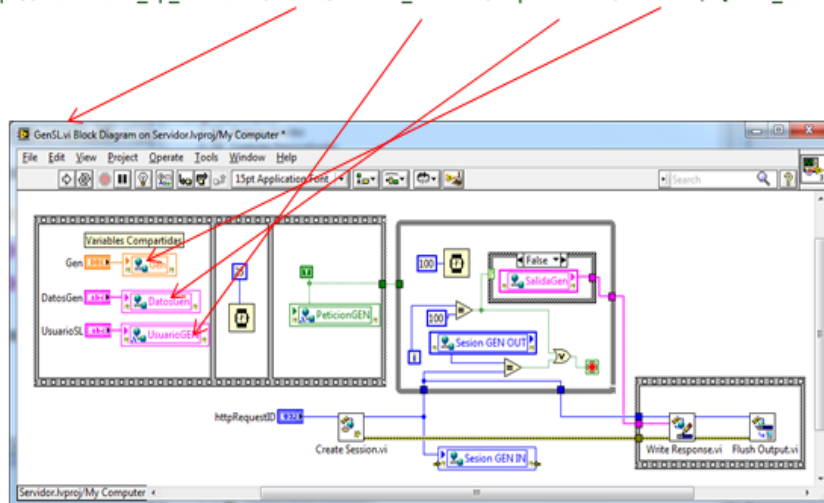
- **Dirección IP o el nombre del equipo:** aquí iría la dirección IP en la cual se encuentre el servicio web.
- **Puerto:** puerto por el que se accede.
- **Nombre del servicio web:** nombre del servicio creado.
- **Nombre del VI:** nombre del VI al que va dirigida la petición. En un mismo servicio web creado en Labview se puede dirigir la petición a diferentes VIs para procesar dicha petición.
- **Terminales de entradas:** datos de entrada al servicio web. En el ejemplo aparecen dos, pero no tienen que ser necesariamente ese número. Cada dato de entrada corresponde con una entrada del VI al que va dirigido.

En la Figura 4.3-36 se muestra una comparativa de la URL de acceso al servicio web con el VI correspondiente. En la imagen se identifican los campos que encaminan los datos a un servicio web u otro.

```

11HTTPRequest("http://direccion_ip_servidor/GenSL/Nombre_Usuario/TipoDatoGen/DatoGen, [HTTP_METHOD, "GET"] , "");

```



**Figura 4.3-36: Comparativa URL de acceso con el VI correspondiente**

Todas las URL's comparten los campos de dir\_IP, puerto y WebService (que es nombre del servicio web). Los dos campos siguientes identifican el VI al que va dirigida la petición (en este caso a GenSL), y el usuario que ha realizado la acción para así guardar la información en la posición de dicho usuario. Los demás campos son propios de cada VI del servicio web y son diferentes en cada uno. Contienen el tipo de dato y los datos que van a recibir.

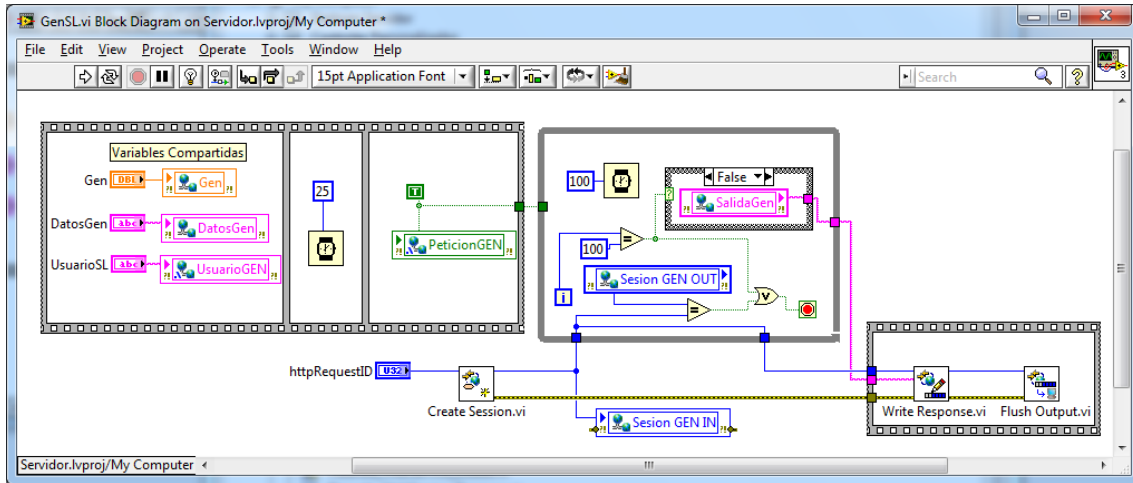
### VI's DEL SERVICIO WEB

En este apartado se va a explicar la estructura de los VI's incluidos en el servicio web. Cabe aclarar que **solo hay un servicio web ejecutándose** que consta de los siguiente VI's:

- **Login:** encargado de recibir nombre y contraseña del usuario que quiere acceder al servidor laboratorio y comenzar a realizar una práctica.
- **Desconexión:** desconecta al usuario cuando éste deja el puesto de laboratorio, borrando su información del sistema.
- **Fuente de alimentación, generador de señales, multímetro y osciloscopio:** atienden las peticiones de configuración de los instrumentos del laboratorio.
- **Placa:** recibe las configuraciones de las placas de prácticas que utiliza el usuario.
- **Estado Servidor:** recibe peticiones que comprueban si el servidor laboratorio está operativo o no, para el indicar a los usuarios que haya en el mundo virtual si es posible o no la interacción con los instrumentos.
- **Reservas:** es un sistema que permite reservar un puesto del laboratorio. Un usuario puede reservar las 24 horas del día y desde el día actual a 6 días después. También está la posibilidad de cancelar una reserva ya hecha.

Cada VI del servicio web es independiente y funciona como una cola FIFO, es decir, si por ejemplo llega una petición de la fuente de alimentación mientras se está procesando la anterior, esta se encola sin perder los datos; si por el contrario llega una petición que no es de la fuente de alimentación, ésta se atiende al instante.

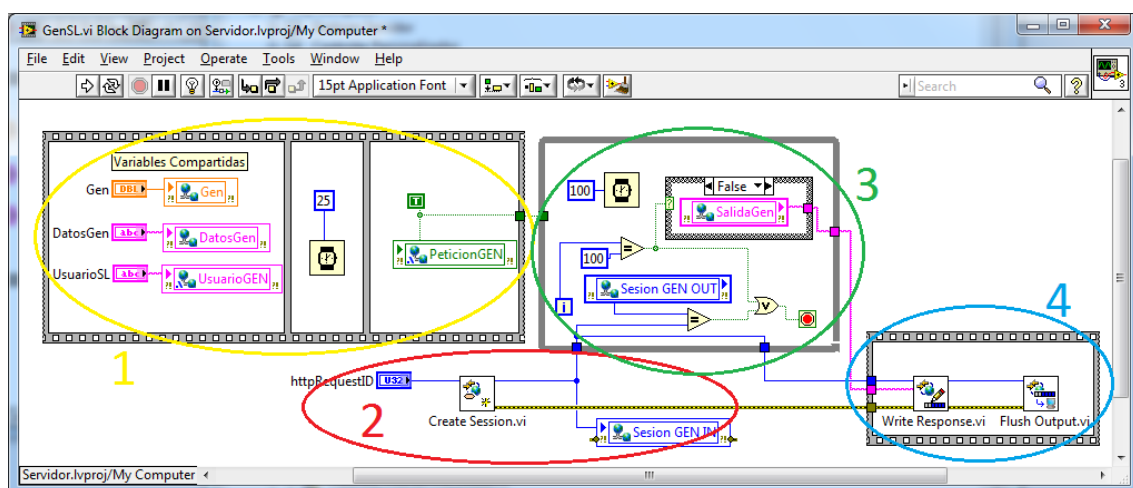
La estructura de los VIs incluidos en el servicio web es la siguiente (Figura 4.3-37):



**Figura 4.3-37: VI servicio web del generador de funciones**

En concreto, se muestra el servicio web que atiende a las peticiones referidas al generador de señales. Los demás tienen una estructura similar, salvo por las variables que utilizan.

El funcionamiento es muy sencillo y se puede dividir en cuatro partes como se muestra en la Figura 4.3-38:



**Figura 4.3-38: Partes del VI del servicio web**



1. Esta primera parte consiste en un “Flat Sequence”, lo que permite ejecutar código de forma secuenciada de izquierda a derecha. Por lo tanto, al llegar una petición se guardan los datos recibidos en las variables compartidas (shared variables en labview<sup>6</sup>), un pequeño retardo y se pone a TRUE una variable compartida, PeticionGEN, que se utiliza como flag para indicar que ha llegado una petición al servicio web del generador. Al modificar esta variable se ejecutaría la parte 3.
2. Con el control “httpRequestID” y el VI “Create Sesion.vi” obtenemos la información (ID) de la petición HTTP entrante que guardamos en la variable compartida “Sesión GEN IN”.
3. Esta parte consta de secuencia while controlada. En cada iteración de la secuencia se comprueba si la variable “Sesión GEN OUT” es igual a “Sesión GEN IN”. La variable “Sesión GEN OUT” se modifica en el servidor laboratorio al terminar todo el proceso. En el momento que tengan el mismo valor, la secuencia while se detiene para enviar la respuesta que se ha generado. También consta de un timer de 10 segundos; si pasado ese tiempo las variables no han sido iguales, se detiene el while y la respuesta que se enviaría sería de error, ya que se entiende que si tarda tanto tiempo se ha producido un error.
4. Con esta parte se envía la respuesta a la petición HTTP mediante el VI “Write response.vi” y a continuación se borra el buffer.

Una vez que se recibe una petición http y se ejecuta el VI correspondiente del servicio web, el proceso de atender dicha petición pasaría al *Servidor Laboratorio*, en concreto, a la pasarela correspondiente a cada VI.

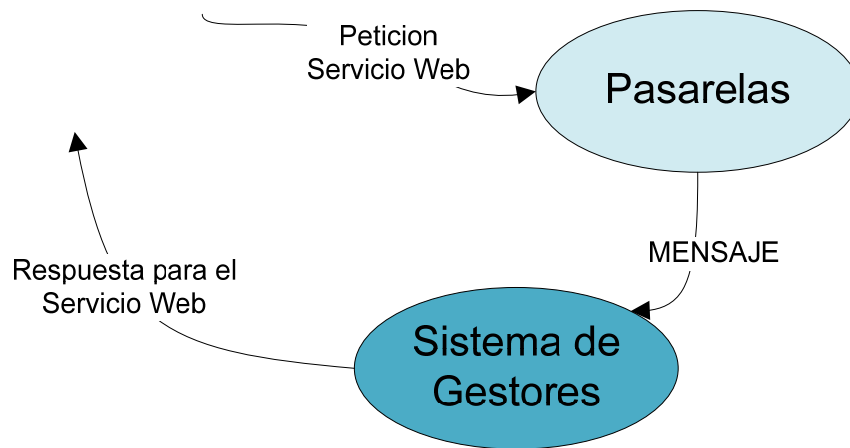
---

<sup>6</sup> Estas variables compartidas es el método que tiene labview para poder pasar un dato recibido por el servicio web a una variable que se pueda usar por VIs independientes al servicio web.

#### 4.3.2.1 *Servidor laboratorio*

El servidor laboratorio es el encargado de procesar la petición recibida en el servicio web con los instrumentos del laboratorio mediante unas pasarelas y un sistema de gestores.

En la Figura 4.3-39 se muestra el camino que seguiría la petición recibida en el VI GenSL.vi del servicio web, para comunicar dicha petición con el servidor de laboratorio, quien la procesa y devuelve la respuesta al VI del servicio web que recibió la petición.



**Figura 4.3-39: Flujograma del Servidor Laboratorio**

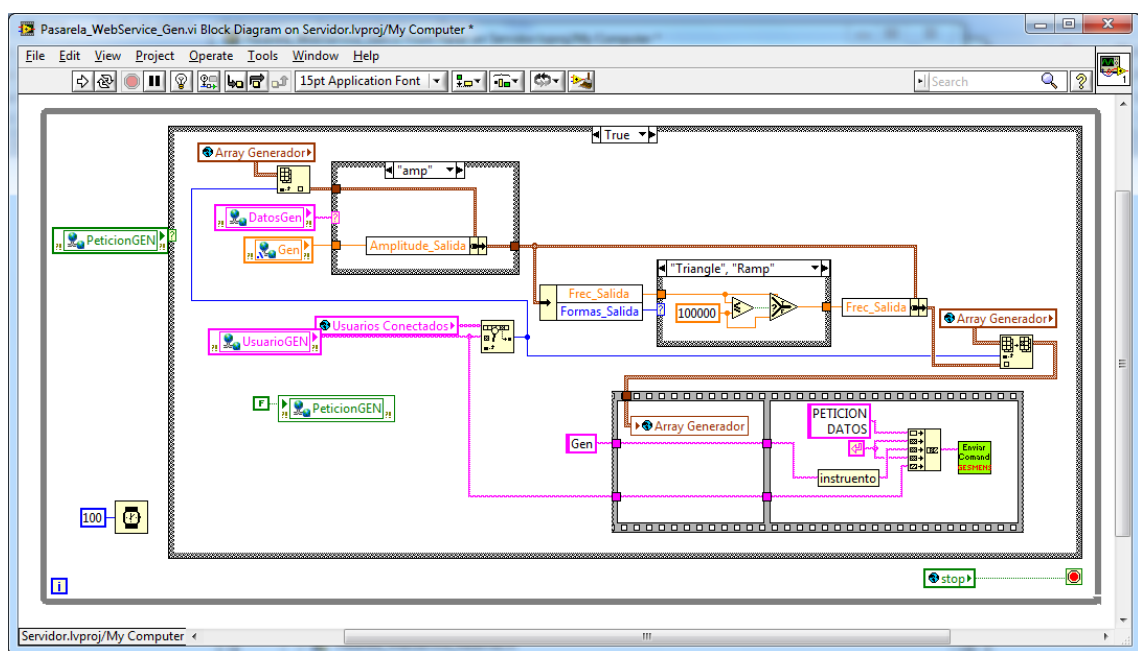
Continuando con el ejemplo considerado en el apartado anterior, referido a una petición al VI del generador del servicio web, cuando se activa el flag *PeticionGEN*, los datos recibidos pasan a ser procesados en la pasarela correspondiente al generador. Estos datos son enlazados entre el VI GenSL.vi y la pasarela del generador mediante las variables compartidas.

A continuación se describe el funcionamiento de las pasarelas y del sistema de gestores.

## PASARELAS DEL SERVICIO WEB

Las pasarelas son VIs que se encargan de enlazar los datos recibidos en la petición HTTP con el servidor laboratorio, para que éste los procese. Hay tantas pasarelas como VIs en el servicio web, aunque todas ellas tienen una estructura similar.

La estructura de las pasarelas se muestra en la Figura 4.3-40:



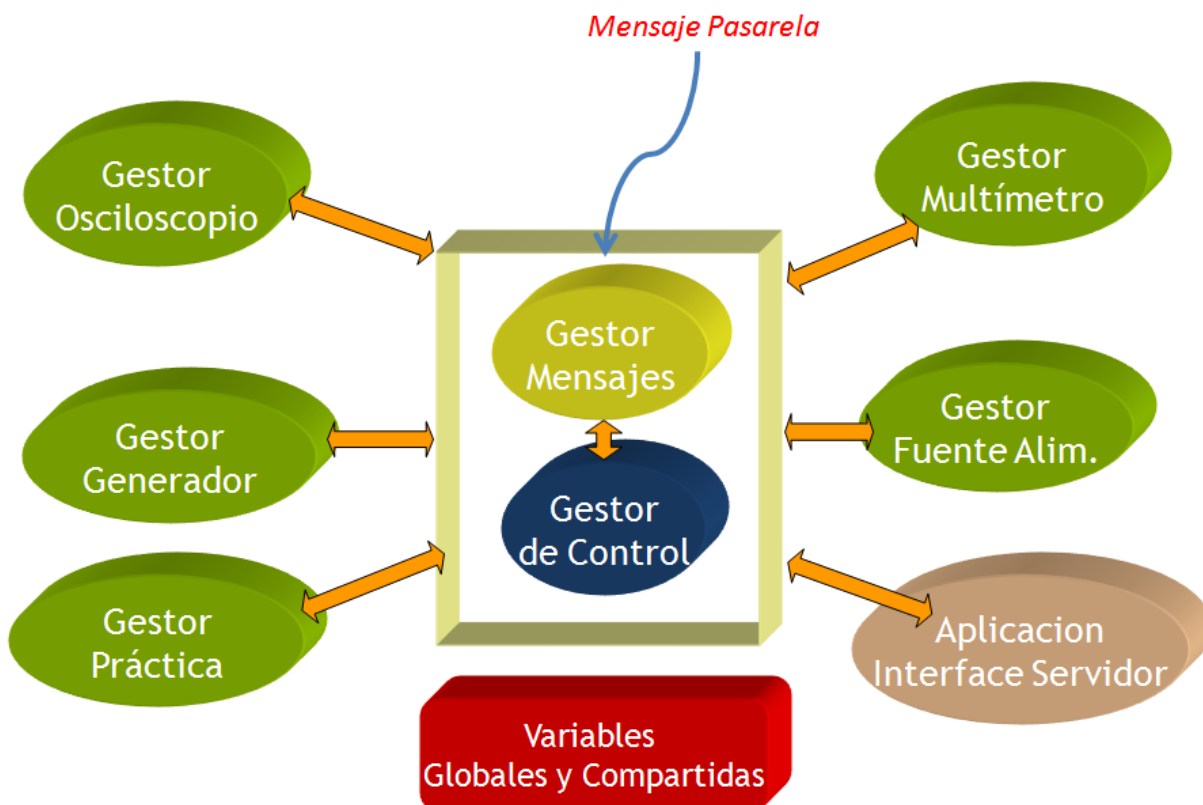
**Figura 4.3-40: Pasarela del servicio web del generador de funciones**

Como se aprecia en la Figura 4.3-40, cada pasarela consiste en una estructura while, para que esté continuamente ejecutándose, y una estructura case de TRUE o FALSE gobernada por la variable compartida "PeticiónGEN". En el caso de que la variable esté a FALSE, este VI no hace nada; en el momento en que pone a TRUE, se ejecuta el código. Básicamente lo que hacen es coger las variables compartidas modificadas en su respectivo VI del servicio web, y con dicha información el servidor laboratorio elabora la respuesta, que será enviada de nuevo por el VI del servicio web que comenzó el proceso. En el momento en que se comienza a ejecutar el código de la pasarela, la variable PeticiónGEN se pone a FALSE, consiguiendo así que el código sólo se ejecute una vez. La última acción que realiza la pasarela es

enviar un mensaje al **Sistema de Gestores** para que comience el proceso de configuración de los instrumentos y así devolver la respuesta a dicha petición.

## SISTEMA DE GESTORES

Este sistema de gestores es el encargado de comunicarse con las pasarelas y gobernar el funcionamiento del sistema hardware. Considerando las funciones que realiza, la estructura es la que se muestra en la Figura 4.3-41:



**Figura 4.3-41: Estructura del sistema de gestores**

La elección de la estructura interna que se utilizará como base para realizar el sistema de gestores es un aspecto clave. El análisis detenido de las especificaciones impone una característica esencial a la estructura que es la **modularidad**. Una estructura modular se puede implementar de diferentes formas, la elegida en nuestro caso, se basará en la utilización de gestores ejecutándose en paralelo, que podrán comunicarse mediante colas y que utilizarán para intercambiar datos variables globales y compartidas. Esta estructura es muy versátil, totalmente

escalable y se puede depurar de forma sencilla. A nivel de código los gestores se implementarán mediante la utilización de una estructura tipo while, incluyendo en su interior una estructura tipo case. A continuación se especificará cuál es la estructura para los gestores y se detallará su implementación con la herramienta de programación LabView.

Antes de entrar en detalle de los gestores implementados, es necesario hacer referencia a la forma de guardar todos los datos relacionados con cada usuario, ya que éstos son utilizados por los gestores para la configuración de los instrumentos de laboratorio.

El sistema consiste en una serie de arrays, inicialmente vacíos, que por cada conexión de un usuario nuevo al servidor laboratorio crea una posición, según el orden de llegada, en cada uno de ellos. Con esto conseguimos que toda la información relacionada con un usuario se encuentre en las mismas posiciones de todos los arrays, siendo fácilmente accesible a todos ellos. El array que se utiliza como índice es el denominado “Array de Usuarios”, que contiene cada nombre de usuario en la posición de llegada, así al llegar nuevos datos de un usuario, se compara con dicho array y se modifican las posiciones correspondientes con ese usuario. Por el contrario, cuando un usuario se desconecta, se borra cada posición relativa a él de todos los arrays. El VI encargado de gestionar todos los arrays es el mostrado en la Figura 4.3-42:

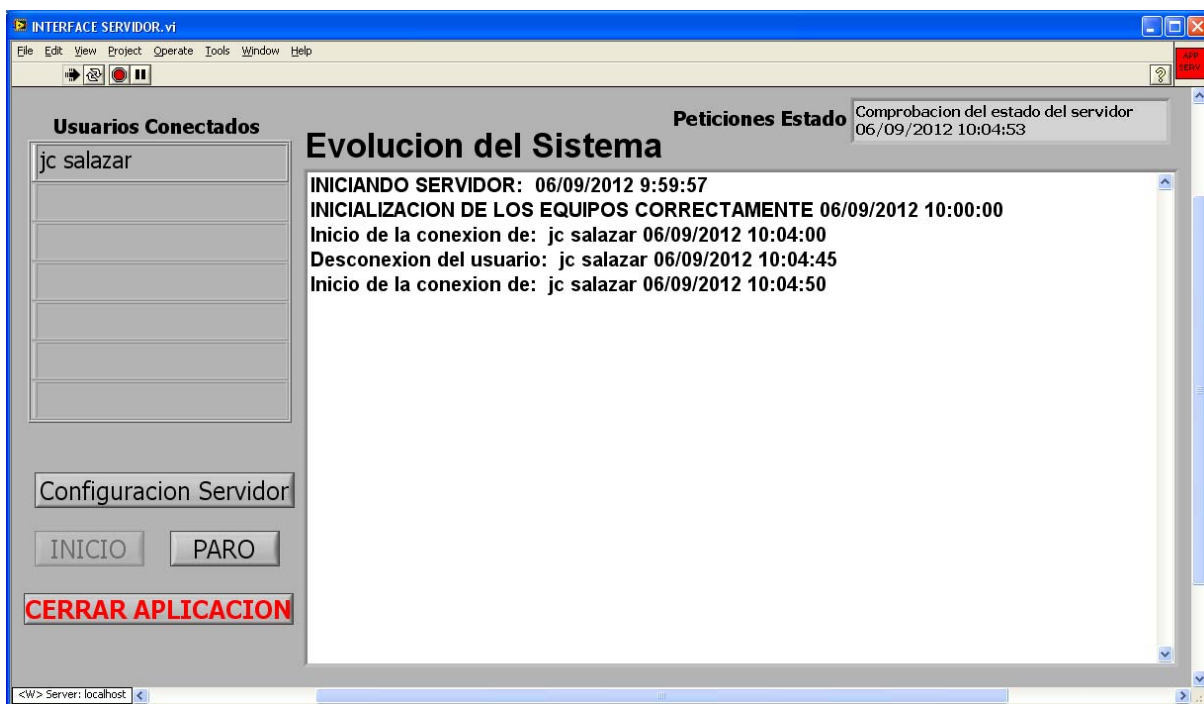


**Figura 4.3-42: VI array instrumentos**

La función y programación de cada uno de los gestores incluidos en la Figura 4.3-41 se indica a continuación:

## **Gestor Interface Servidor**

Este VI también tiene la estructura de gestor explicada, pero además realiza otras funciones. Es el encargado de lanzar todos los gestores, sub VIs y pasarelas del servicio web incluidos en el servidor laboratorio; recoge los mensajes informativos de la evolución del servidor, enviados por los demás gestores y los representa en un historial, por lo que hace de interface de usuario para la depuración de esta parte del laboratorio remoto. En la Figura 4.3-43 se muestra su apariencia:



**Figura 4.3-43: Interface Servidor**

En la interface de este gestor se diferencian tres partes:

1. Usuarios conectados al servidor laboratorio: muestra los usuarios que han cogido un puesto de laboratorio.
2. Historial de las acciones que realizan los usuarios (Log): representa la evolución de las acciones realizadas por los usuarios. Cada acción queda registrada con fecha, hora y el nombre del usuario que realiza dicha acción. Toda la información de este historial se guarda en un fichero, para el control y evaluación de las acciones de los usuarios.

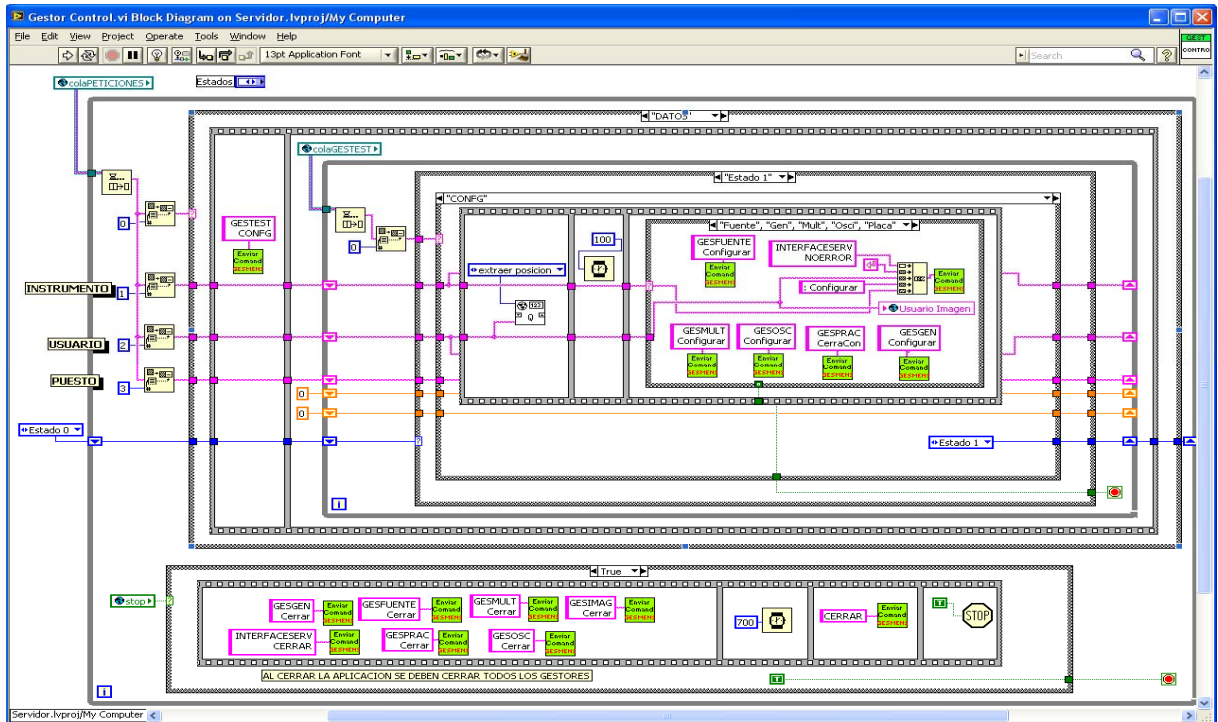
3. Botones de configuración del servidor (inicio, paro, configuración inicial, cerrar aplicación): esta es la parte del interface de usuario, con la que se puede controlar la aplicación servidor laboratorio.

En el apartado 6 *MANUAL DE USUARIO II: SERVIDOR LABORATORIO* se explicará el funcionamiento de este programa.

### Gestor de Control

Controla el funcionamiento global de la aplicación. Está realizado como un autómata de estados y se encarga de enviar órdenes al resto de los gestores, indicando las acciones que en cada momento éstos deben realizar. Así mismo, el Gestor de control recibe del resto de los gestores información sobre los resultados de la ejecución de las acciones realizadas por éstos (realización de una medida, error en una comunicación con un equipo, fin de un establecimiento de señal, etc.). Es el único VI del sistema que posee capacidad de decisión.

En la Figura 4.3-44 se muestra parte del código que implementa este gestor.



**Figura 4.3-44: Código del Gestor de Control**

Este programa se ejecuta cada vez que se recibe un mensaje de alguna de las pasarelas del servicio web indicando que el usuario remoto ha realizado algún tipo de petición. A partir de ese momento comenzaría a ejecutarse un autómata que a través de mensajes a los gestores que correspondan, completaría las acciones que fueran necesarias para atender la petición http producida por una acción del usuario remoto.

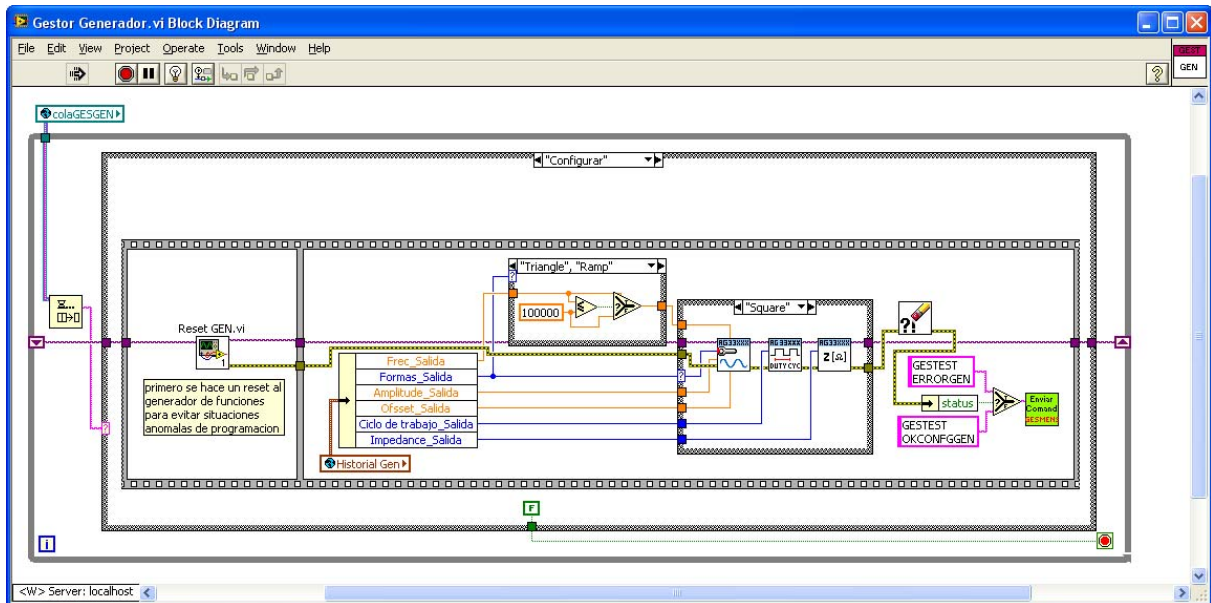
La utilización de dos colas de mensajes en este gestor es muy importante ya que es necesario que se completen todas las acciones que se requieran para atender la petición de un usuario antes de poder atender a una nueva petición de otro usuario o del mismo. Anteriormente se comentó que VIs pertenecientes al servicio web son independientes y que si llegan peticiones a dos diferentes, éstas se atienden, sin necesidad de que termine la primera en ser atendida. Pues bien, si ocurre esto, en el gestor de control se quedan encoladas las peticiones, ya que por cada petición es necesario que terminen todas las acciones y devuelva la respuesta a la petición antes de atender otra.

### **Gestor Prácticas, Gestor Multímetro, Gestor Fuente Alimentación, Gestor Generador y Gestor Osciloscopio**

Se encargarían de realizar las operaciones apropiadas, indicadas por el Gestor de Control, sobre el sistema hardware definido en el apartado 3.3 *Sistema hardware*. Los Gestores de los instrumentos controlan cada uno de los instrumentos del laboratorio y el Gestor Prácticas se encarga de configurar los circuitos que forman parte de cada práctica.

Todos estos gestores poseen la misma estructura y se implementan de la misma forma. En la Figura 4.3-45 se muestra, a modo de ejemplo, el código que implementa el Gestor Generador.





**Figura 4.3-45: Código del Gestor Generador**

Este programa se ejecuta cada vez que se reciba un mensaje procedente del Gestor de Control. Se realiza la acción sobre el instrumento correspondiente o sobre el sistema hardware del bloque Experimento y se avisa al Gestor de Control de la finalización de dicha acción.

Para poder controlar los instrumentos es necesario utilizar una serie de drivers proporcionados por los fabricantes de los instrumentos. Estos drivers contienen un conjunto de funciones que simplifican la programación de los dispositivos sin necesidad de conocer los comandos específicos de los mismos.

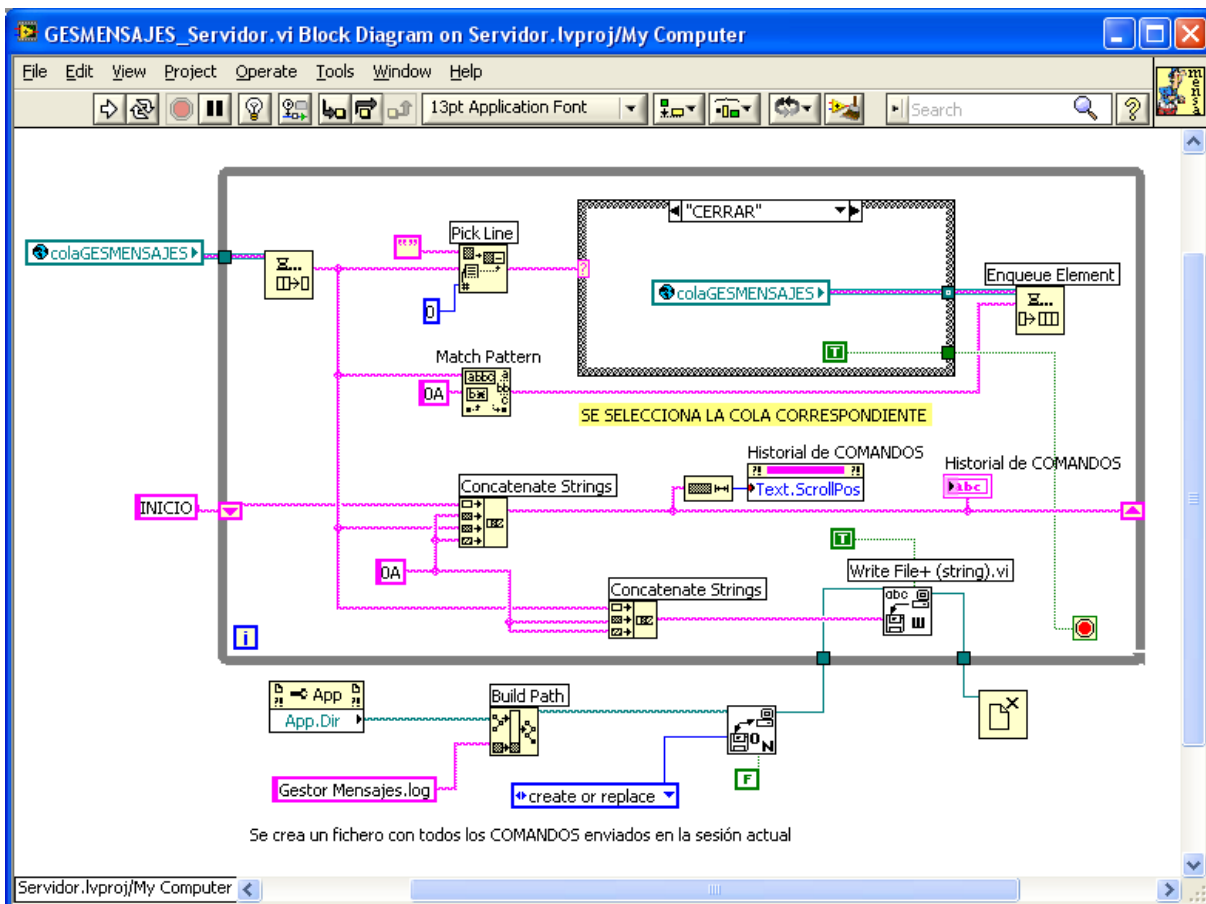
Al igual que para controlar los instrumentos, para el manejo del hardware del bloque Experimento es necesario utilizar el driver, desarrollado a la vez que se implementó el hardware. Dicho driver es utilizado en el gestor de prácticas.

### **Gestor Mensajes Servidor**

Su misión es la de encaminar los mensajes de comunicación entre los gestores y ofrecer la posibilidad de consultar una variable que contiene el historial con todas las acciones realizadas en la aplicación servidor, facilitando la depuración.

Éste VI es el encargado de comunicar el mensaje enviado desde alguna de las pasarelas con el gestor de control para que comience el proceso de configuración del sistema hardware y generar la respuesta.

En la Figura 4.3-46 se muestra el código que implementa este gestor:

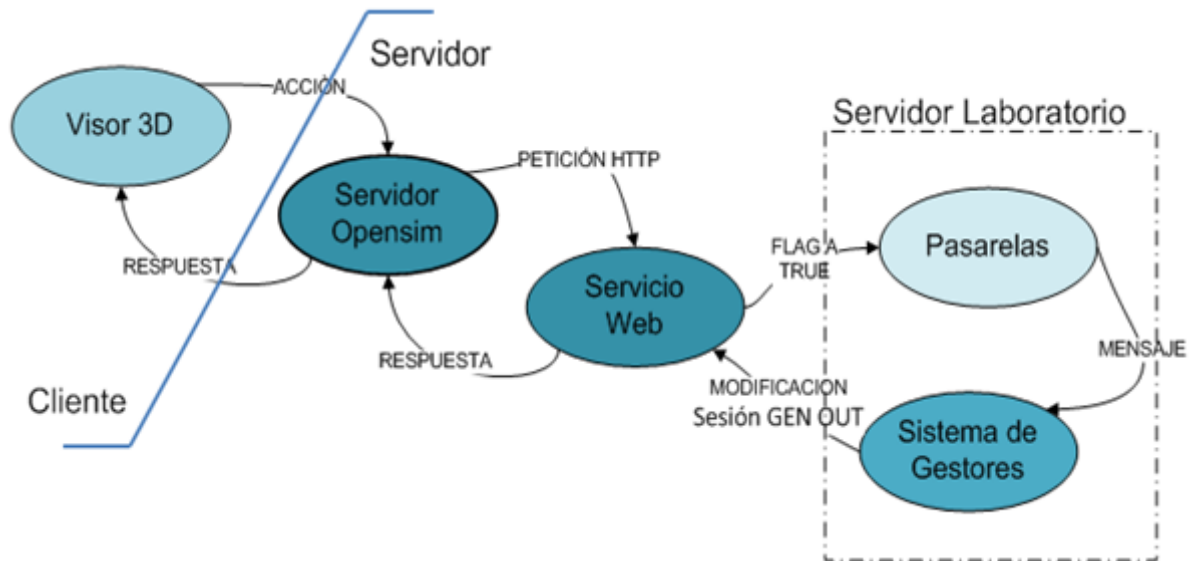


**Figura 4.3-46: Código del Gestor de Mensajes**

Este programa se ejecuta cada vez que se recibe un mensaje de un gestor o de una pasarela y se limita a redirigir el contenido de dicho mensaje, asignándolo a la cola del gestor que corresponda, y a almacenar, a modo de historial, el contenido de los mensajes en una variable.

## 4.4 Visión global del sistema software

A modo de resumen, y para aclarar el proceso de comunicación desde que un usuario realiza una acción en el mundo virtual, hasta que se recibe la respuesta a dicha acción, en la Figura 4.4-1 se muestra el flujograma completo:



**Figura 4.4-1: Flujograma del sistema software**

Este flujograma representa de forma completa el proceso completo entre las acciones del usuario y como se genera su respuesta. Cada una de las partes diferenciadas ha sido explicada en los diferentes apartados anteriores.



# 5

## MANUAL DE USUARIO I: VISOR 3D



## 5.1 Introducción

Este manual está dirigido a aquellos usuarios que deseen disponer de una herramienta de aprendizaje, a través de Internet, que permita adquirir o repasar conocimientos prácticos relacionados con el área de la electrónica.

Con la ayuda de este manual se pretende que el usuario conozca las posibilidades de interacción que proporciona el laboratorio eLab-3D, un entorno que emula un Laboratorio de Electrónica en tres dimensiones donde se podrá operar de forma real con los instrumentos típicos de un laboratorio (fuente de alimentación, generador de funciones, multímetro y osciloscopio) para comprobar y verificar el funcionamiento de diferentes circuitos electrónicos.

El manual se ha estructurado con dos objetivos básicos: facilitar la comprensión del funcionamiento básico del laboratorio y permitir al usuario evaluar algunas de las posibilidades de aprendizaje que ofrece la plataforma. Incluye dos partes bien diferenciadas, una descripción inicial básica del entorno y un tutorial guiado que incluye una serie de actividades prácticas. Dichas actividades se centrarán en el manejo de los instrumentos del laboratorio y en la adecuada configuración de un sencillo circuito electrónico, un divisor de tensión resistivo, sobre el que se realizarán una serie de medidas eléctricas que permitirán comprobar su funcionamiento bajo diferentes condiciones de excitación.

Como recomendación, antes de comenzar a leer este manual y para formarse una idea inicial sobre el laboratorio eLab-3D, le aconsejamos visionar el vídeo disponible en el siguiente enlace:

<http://innovacioneducativa.upm.es/laboratoriosvirtuales/node/42>

Por último, es necesario considerar los siguientes aspectos al trabajar con el laboratorio:

- Para utilizar el laboratorio eLab-3D no se requieren conocimientos técnicos sobre las tecnologías que se han empleado para desarrollar la plataforma.
- La interacción con el entorno 3D es dependiente de factores de difícil control como son: las prestaciones del ordenador del usuario, sobre todo las de la tarjeta gráfica que tenga instalada, la configuración del antivirus utilizado o el ancho de banda de la conexión a Internet disponible. Asimismo hay que tener en cuenta que la versión del servidor Opensim con la que se ha desarrollado el laboratorio no es totalmente estable. Lamentablemente estos factores pueden influir negativamente en la experiencia con el laboratorio.

## 5.2 Visor 3D

A continuación se indicará cómo es el proceso de instalación del visor Firestorm y se comentarán, de forma breve, algunas nociones básicas sobre su manejo.

### 5.2.1 Instalación

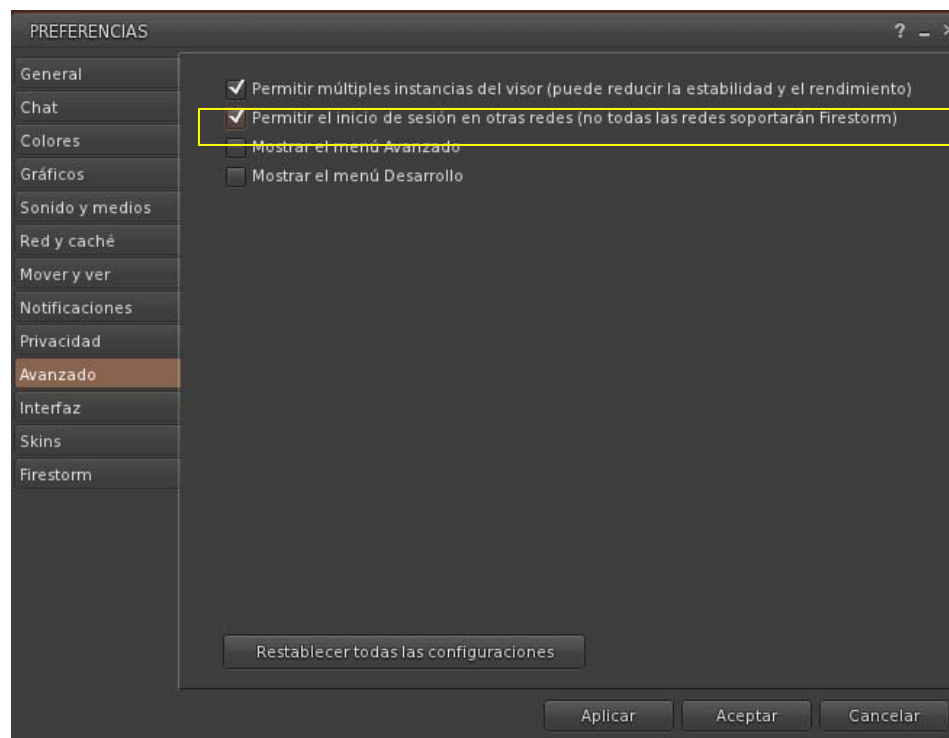
Para instalar el visor Firestorm se deben seguir las siguientes instrucciones:

- Descargar la versión del visor Firestorm (no descargar Phoenix) de la página: <http://www.phoenixviewer.com/downloads.php>
- Instalar el programa aceptando las opciones que aparezcan por defecto.
- Configurar el acceso al servidor del laboratorio. Es necesario copiar el fichero <grids.fallback.xml>, que se le ha facilitado, en la carpeta <app\_settings>, ubicada en el directorio donde Firestorm se ha instalado (por defecto: C:\Archivos de programa\Firestorm-Release\app\_settings).
- Ejecutar el programa. Aparecerá una ventana como la que se muestra en la Figura 5.2-1



**Figura 5.2-1: Visor Firestorm 1**

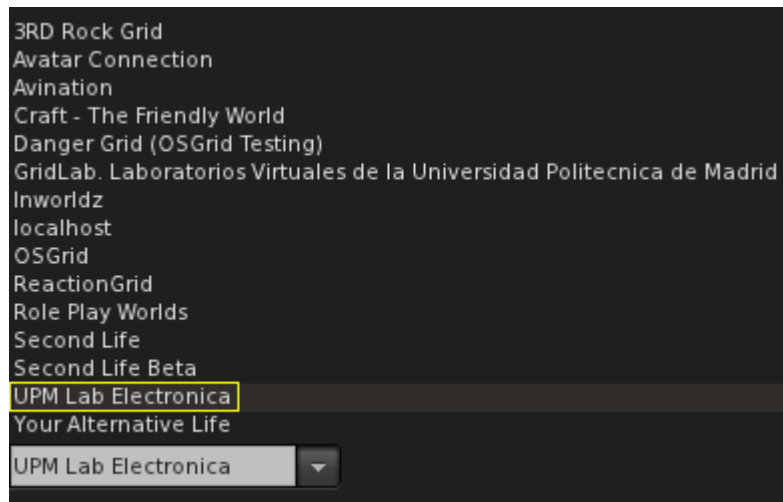
- Acceder al menú Visor → Preferencias → Avanzado y seleccionar la opción “Permitir el inicio de sesión en otras redes” (ver Figura 5.2-2), finalmente pulsar Aceptar.

**Figura 5.2-2: Visor Firestorm 2**

- En la parte inferior del visor aparecerán las opciones para entrar en el mundo virtual (ver Figura 5.2-3). Desplegar el menú de la derecha “Iniciar en la red:” y seleccionar “UPM Lab Electrónica” (Figura 5.2-4).



**Figura 5.2-3: Visor Firestorm 3**



**Figura 5.2-4: Visor Firestorm 4**

- Introducir el nombre de usuario y la contraseña que se le haya asignado y pulsar en “Iniciar sesión”
- Aparecerá su avatar (por defecto de género femenino) en la posición “Mi Base” (exterior del edificio Laboratorio de Electrónica)<sup>7</sup>. En posteriores accesos, por defecto, aparecerá en la última posición en la que estaba cuando abandonó el mundo virtual. En el desplegable “Iniciar en” puede cambiar esta opción.

## 5.2.2 Manejo básico del visor

A continuación se proporcionan algunas indicaciones básicas para facilitar el manejo del visor. Aunque es muy intuitivo, se recomienda hacer pruebas sencillas

<sup>7</sup> Si aparece alguna ventana para que active la posibilidad de visualizar videos, por defecto, permita dicha opción.

de las opciones que se comentan a continuación para familiarizarse lo antes posible con el manejo del mismo.

- **Cambiar apariencia del avatar:** por defecto la apariencia del avatar puede no ser al gusto de todos, es femenino y lleva una camiseta morada y unos pantalones rojos. Por ello, en el mundo virtual se dispone de una serie de opciones de personalización. Si desea hacer algún cambio en la apariencia física del avatar puede acceder a los siguientes menús: Avatar → Mi apariencia, y dentro de la nueva ventana que se abrirá, en la pestaña “PUESTO” podremos editar cada una de las partes del avatar, se puede cambiar entre hombre y mujer y modificar todo lo relacionado con la apariencia física y la ropa que lleva el avatar.



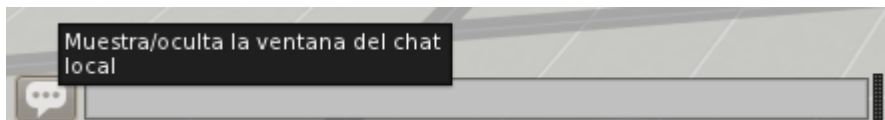
**Figura 5.2-5: Avatar editando apariencia**

- Desplazarse por el mundo:
  - Puede mover su avatar usando las teclas de dirección de su teclado o mediante el cuadro “Controles de Movimiento”, que se encuentra en el menú Avatar → Controles de Movimiento (Figura 5.2-6).



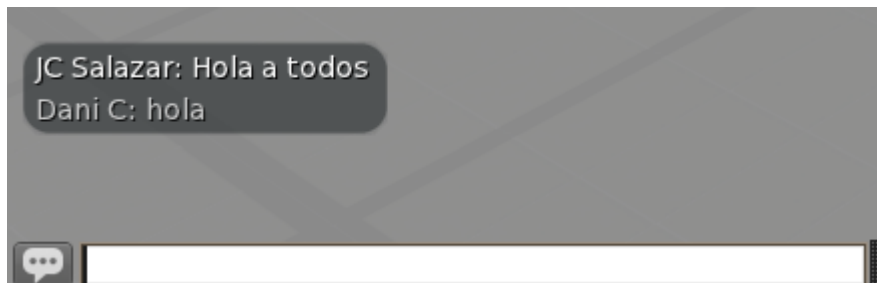
**Figura 5.2-6: Controles de movimiento**

- Puede hacer volar su avatar haciendo clic en el icono ubicado en la parte inferior derecha del cuadro de “Controles de Movimiento”. También puede hacer saltar y volar su avatar con la tecla RePág. Manteniéndola pulsada el avatar asciende para volar (si se pulsa sólo una vez da un salto), y con AvPág el avatar desciende. Para desactivar el vuelo pulsar “Dejar de volar” o la tecla Inicio.
- **Comunicarse:** en el visor hay dos formas de comunicación, con el chat público o privado, y mediante voz (sólo en algunos visores). El registro del chat se puede mostrar/ocultar al pulsar el icono situado a la izquierda de la barra del chat (Figura 5.2-7).



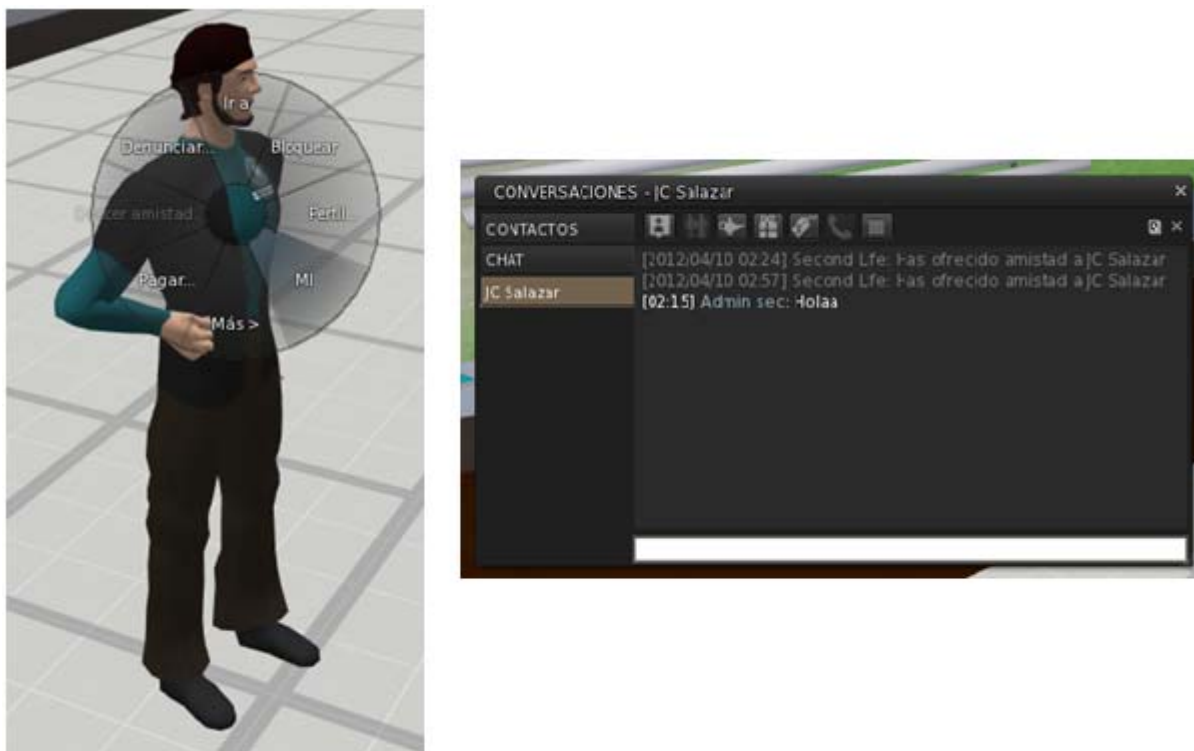
**Figura 5.2-7: Registro del chat**

- Para comunicarse con el chat público, una vez habilitado, solo hay que escribir el mensaje deseado en el espacio reservado para la barra del chat. Cualquier mensaje enviado por este método será público para todos los avatares presentes en el mundo virtual dentro de un rango espacial determinado (aproximadamente unos 20 metros). Los mensajes de comunicación aparecerán en una ventana como se muestra en la Figura 5.2-8.



**Figura 5.2-8: Mensajes del chat público**

- Para comunicarse con un avatar por un chat privado puede hacer clic con el segundo botón del ratón sobre el avatar con el que desea comunicarse y seleccionar “MI” (ver Figura 5.2-9 izquierda), aparecerá una ventana de chat privado como se muestra en la ver Figura 5.2-9 derecha.



**Figura 5.2-9: Mensaje por chat privado**

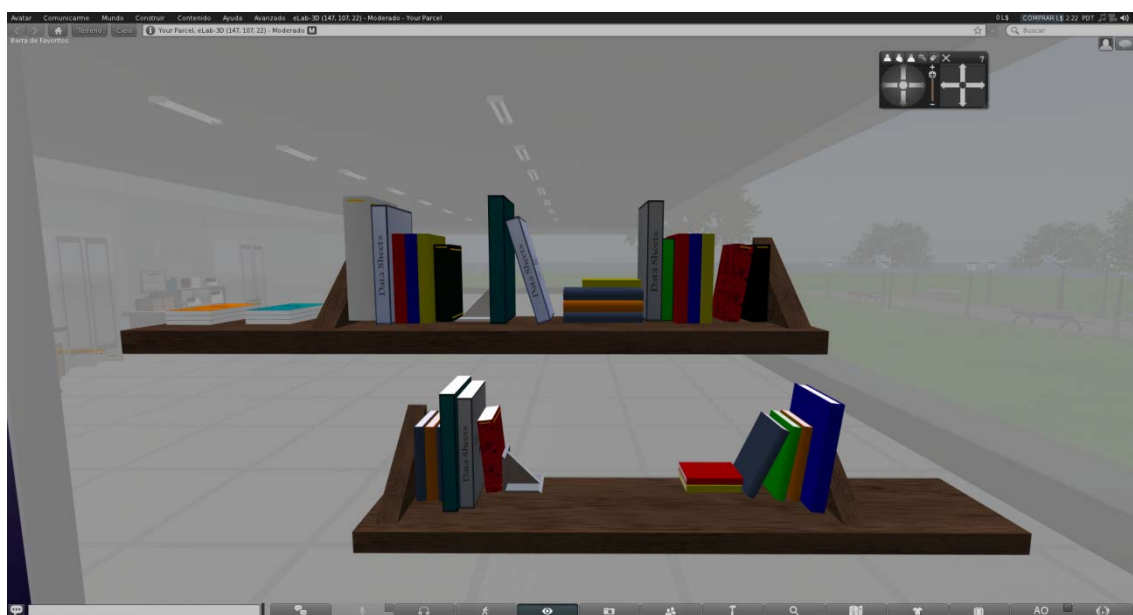
- **Cámara:** para controlar la cámara que permite la visualización de los objetos del mundo virtual se tienen las siguientes opciones.
  - Usar la rueda del ratón para alejarse o acercarse.
  - Mantener pulsada la tecla Alt y hacer clic sobre el punto al que queremos desplazar la cámara.
  - Mantener pulsada la tecla Alt y a la vez mantener pulsado el botón izquierdo del ratón para conseguir un desplazamiento continuo de la cámara según se mueva el ratón.
  - Usar el cuadro de “Control de Cámara” que se encuentra en el menú Avatar → Controles de la Cámara (Figura 5.2-10).



**Figura 5.2-10: Controles de cámara**

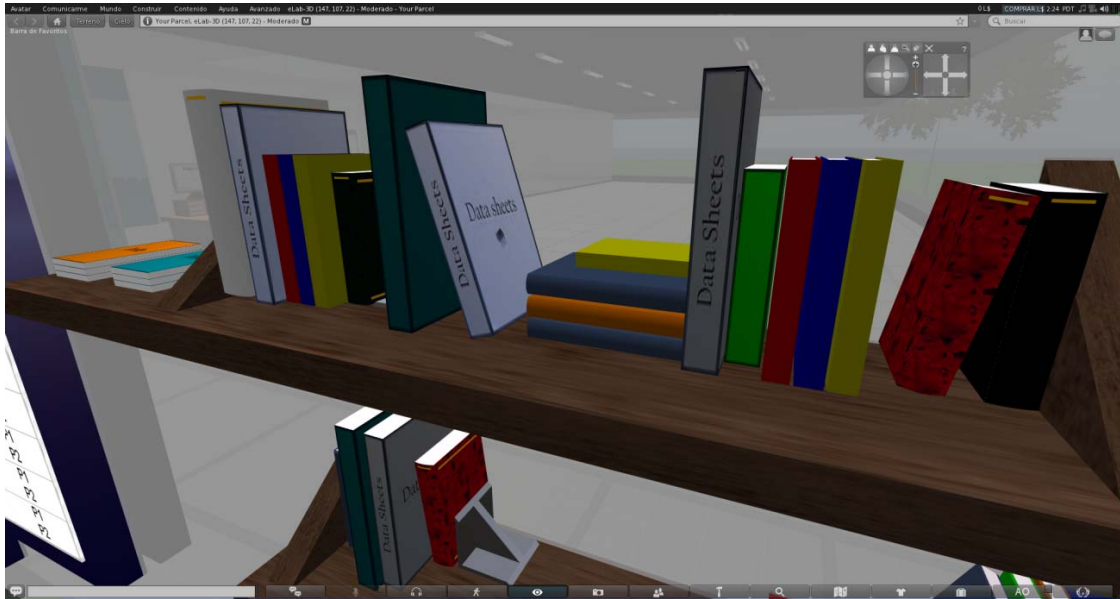
Es muy importante realizar varias pruebas con el manejo de la cámara ya que su dominio será clave para efectuar con facilidad las diferentes actividades prácticas. Se propone el siguiente entrenamiento:

- Entre al hall del edificio y enfoque la cámara a las estanterías ubicadas a la derecha de la entrada, como se muestra en la Figura 5.2-11.



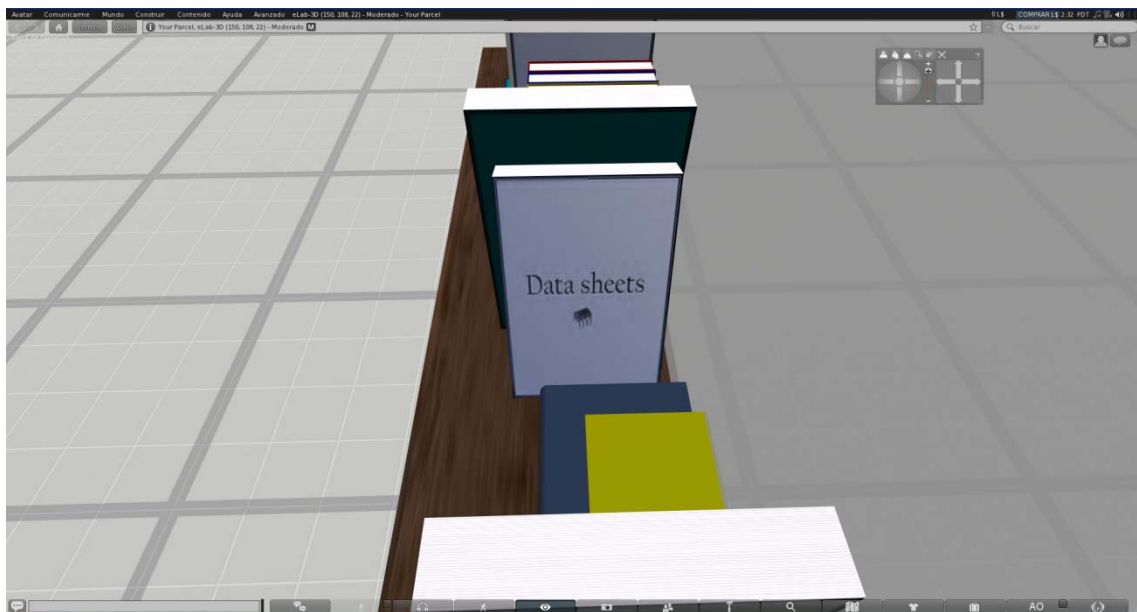
**Figura 5.2-11: Imagen de entrenamiento 1**

- Manipule la cámara hasta conseguir la imagen que se muestra en la Figura 5.2-12.



**Figura 5.2-12: Imagen de entrenamiento 2**

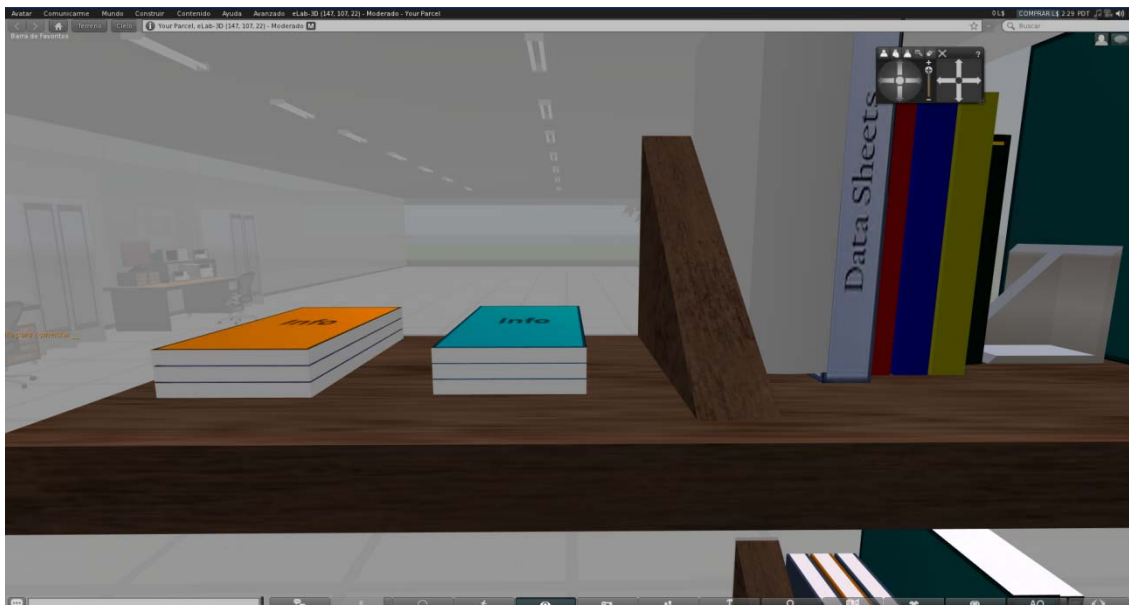
- Manipule la cámara hasta conseguir la imagen que se muestra en la Figura 5.2-13.



**Figura 5.2-13: Imagen de entrenamiento 3**

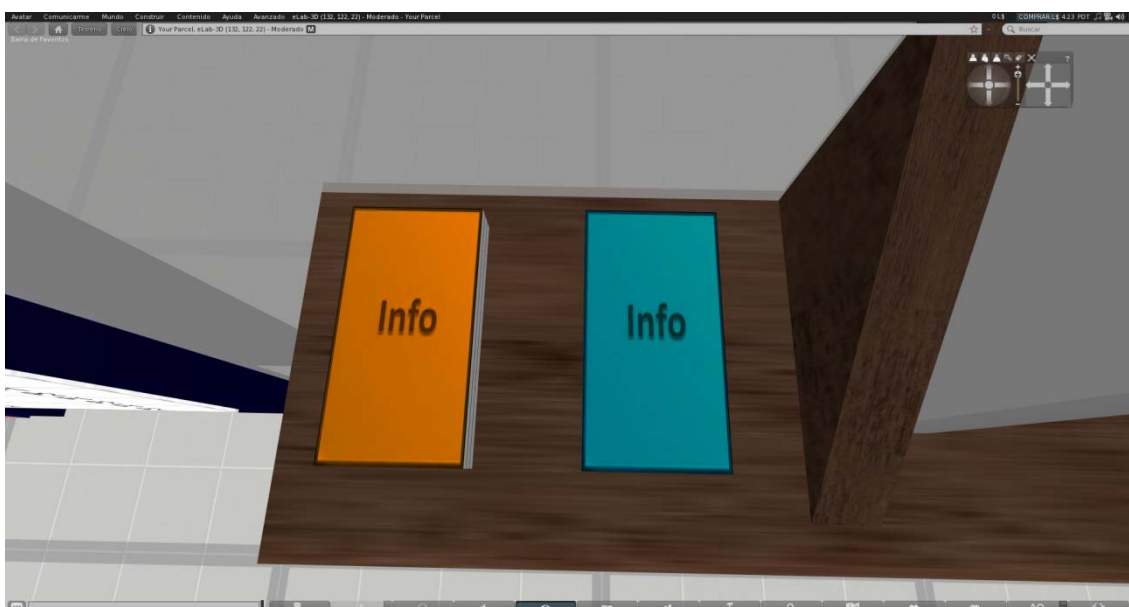
- Manipule la cámara hasta conseguir la imagen que se muestra en la Figura 5.2-14.





**Figura 5.2-14: Imagen de entrenamiento 4**

- Manipule la cámara hasta conseguir la imagen que se muestra en la Figura 5.2-15.



**Figura 5.2-15: Imagen de entrenamiento 5**

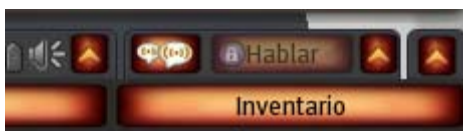


- **Interactuar con los objetos:** hacer clic con el botón derecho del ratón para mostrar un menú contextual que nos permitirá realizar acciones con determinados objetos. Por ejemplo, pruebe a sentarse en uno de los sillones situados al final del hall (Figura 5.2-16). Cuando se trabaje en el laboratorio, como comprobará más adelante, la interacción con la mayoría de los objetos será más sencilla, ya que sólo será necesario hacer clic con el ratón (tocar) en el objeto para que éste realice la acción definida en el script (código) del objeto.



**Figura 5.2-16: Sala de estar**

- **Inventario:** muestra los objetos, texturas... que posee nuestro avatar. Para que se muestre el inventario hay dos opciones, mediante el icono con forma de maleta que hay en la barra inferior (Figura 5.2-17) o en el menú Ver → Inventario:



**Figura 5.2-17: Botón para abrir el inventario**

### 5.2.3 Otras opciones del visor

El visor dispone de múltiples opciones adicionales que no van a ser definidas en este manual. A continuación se indican algunas otras opciones a tener en cuenta:

- **Cambiar hora del día:** se puede elegir el periodo del día del mundo virtual en el menú Mundo → Posición del sol → Mediodía (mayor visibilidad).
- **Menú Avatar → Preferencias:** se disponen en este menú de múltiples opciones relacionadas con la configuración del visor. Respecto al menú “Gráficos” es necesario tener configurada la opción “Calidad de dibujo” como mínimo en “Alta” para poder ver ciertos efectos de los objetos. Esto puede ralentizar el visor dependiendo de la tarjeta gráfica de su ordenador.
- **Desactivar información adicional:** se puede hacer desaparecer la información que aparece al situar el cursor encima de un objeto del entorno en el menú Avatar → Preferencias → Interfaz, desmarcando la casilla “Básicas” dentro de la opción “Bulas”.

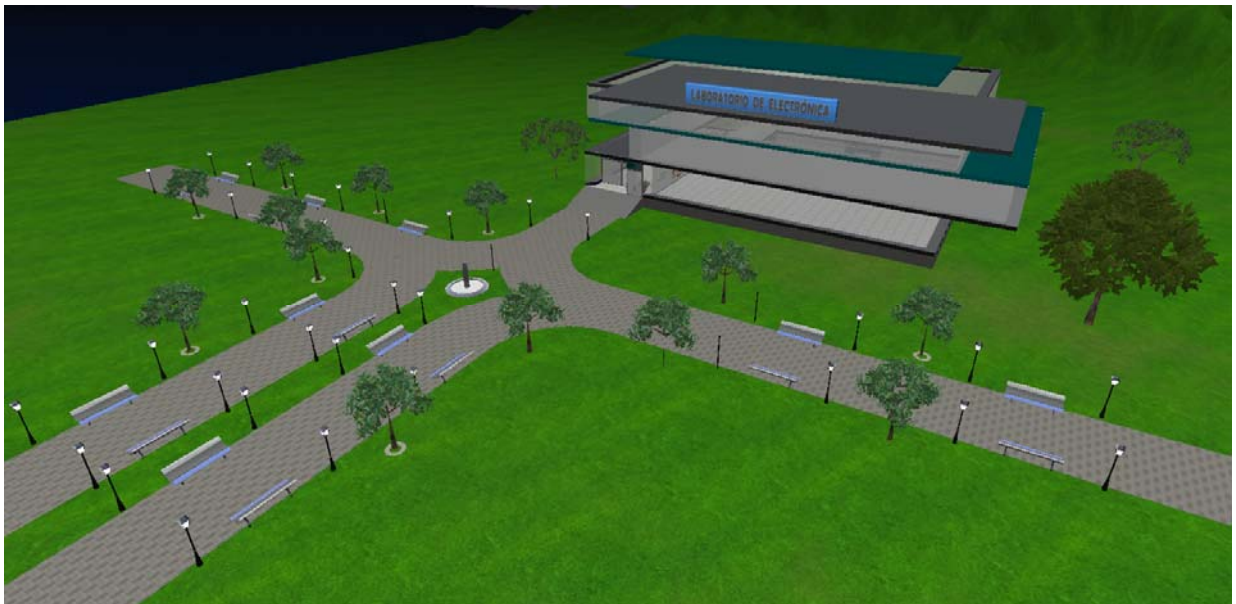
## 5.3 Tutorial de uso del laboratorio eLab-3D

Para facilitar una rápida comprensión de las posibilidades que ofrece el laboratorio eLab-3D se ha preparado este tutorial guiado en el que se incluyen una serie de acciones que el usuario debe realizar justo después de los comentarios explicativos. Según avance en la ejecución de este tutorial es posible que detecte algún error de funcionamiento, la mayoría de ellos son achacables fundamentalmente a la inestabilidad de la versión del servidor Opensim, al visor utilizado, a las prestaciones de su ordenador (tarjeta gráfica) o a la red de acceso a Internet. Cuando observe algún tipo de error, independientemente de cuál haya sido la causa, por favor, le rogamos nos lo comunique por correo electrónico.

El laboratorio de electrónica está construido en una región o isla que incluye el edificio principal del laboratorio, donde se encontrarán los puestos de laboratorio a utilizar por los usuarios, el terreno que rodea al edificio y un pequeño parque de esparcimiento. A continuación se realizan breves descripciones de cada uno de los componentes mencionados.

### 5.3.1 Isla

La isla o región incluye todo el escenario que constituye el laboratorio. Una imagen de la misma se muestra en la Figura 5.3-1:



**Figura 5.3-1: Vista aérea de la isla**

**Acción 1:** con la tecla Alt pulsada y moviendo la rueda del ratón, mover la cámara hasta ver la isla desde arriba, más o menos como se ve en la Figura 5.3-1.

### 5.3.2 Parque de esparcimiento

Lugar en los que los diferentes usuarios pueden pasear o sentarse en los bancos disponibles para chatear e intercambiar impresiones (Figura 5.3-2).



**Figura 5.3-2: Vista del exterior del edificio**

**Acción 2:** mover el avatar por el parque y sentarlo en un banco.

### 5.3.3 Edificio

Ubicado en el centro de la isla se encuentra el edificio que alberga el laboratorio (Figura 5.3-3).



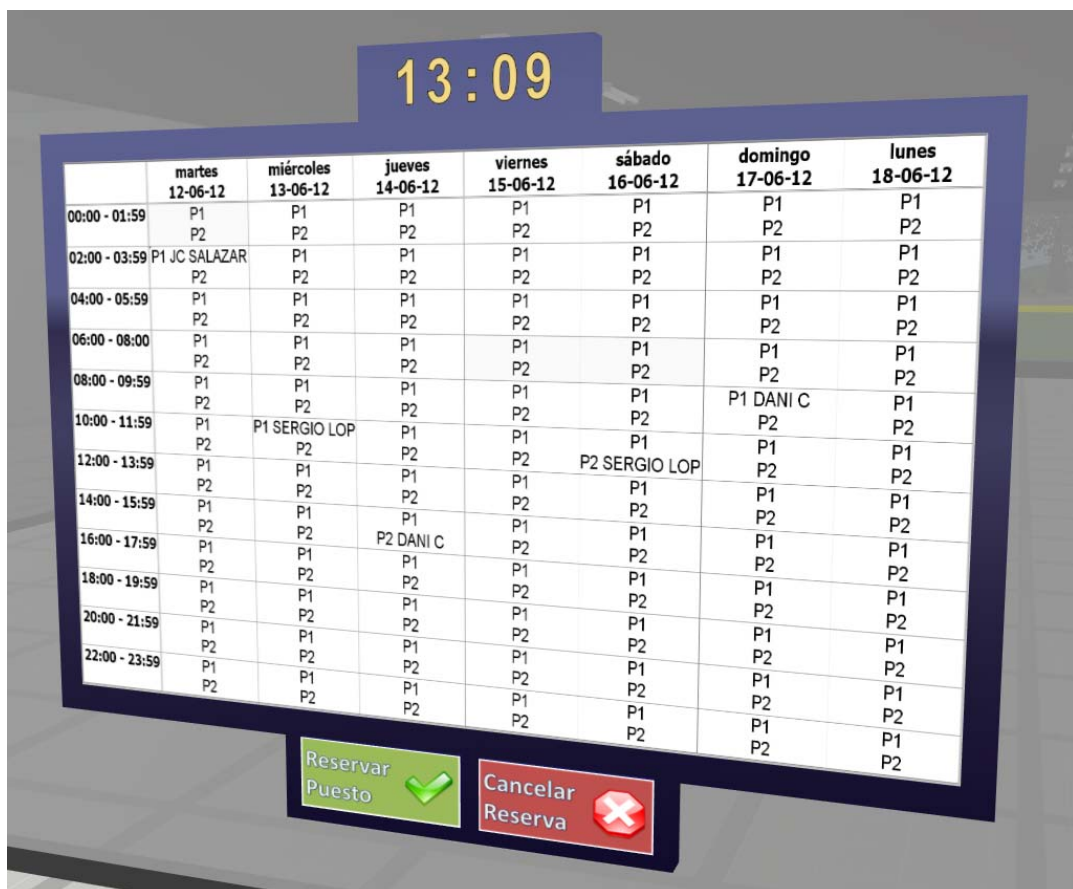
**Figura 5.3-3: Edificio de electrónica**

Inicialmente el edificio consta de un hall y tres salas principales:

- **Hall**



Al entrar en el edificio existe un vestíbulo donde se pueden observar los siguientes elementos:

- Un panel informativo con la ocupación de los puestos de laboratorio y un botón de reservas para poder formalizar la reserva de un puesto de laboratorio (Figura 5.3-4). Cuando se toca dicho botón aparecen una serie de menús que permiten seleccionar el día, hora y puesto de laboratorio. Las reservas de los puestos de laboratorio siempre se realizarán por periodos de dos horas. También existe un botón de cancelación de reservas. El sistema de reservas avisa al usuario cuando faltan diez, cinco y un minuto para terminar la sesión de dos horas reservada.



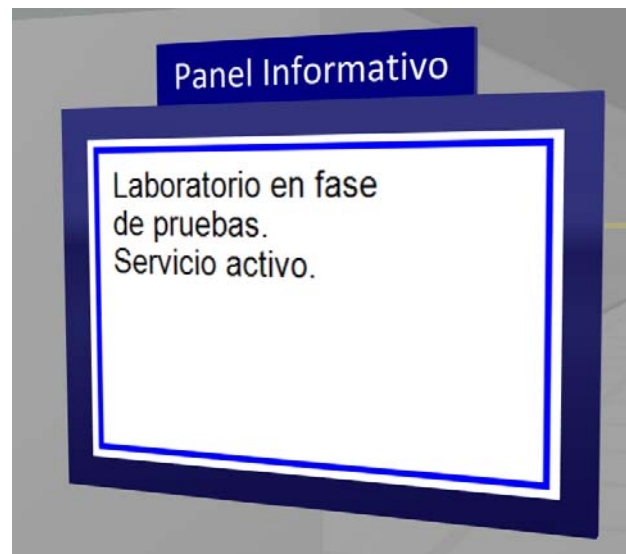
13:09

|               | martes<br>12-06-12  | miércoles<br>13-06-12 | jueves<br>14-06-12 | viernes<br>15-06-12 | sábado<br>16-06-12  | domingo<br>17-06-12 | lunes<br>18-06-12 |
|---------------|---------------------|-----------------------|--------------------|---------------------|---------------------|---------------------|-------------------|
| 00:00 - 01:59 | P1<br>P2            | P1<br>P2              | P1<br>P2           | P1<br>P2            | P1<br>P2            | P1<br>P2            | P1<br>P2          |
| 02:00 - 03:59 | P1 JC SALAZAR<br>P2 | P1<br>P2              | P1<br>P2           | P1<br>P2            | P1<br>P2            | P1<br>P2            | P1<br>P2          |
| 04:00 - 05:59 | P1<br>P2            | P1<br>P2              | P1<br>P2           | P1<br>P2            | P1<br>P2            | P1<br>P2            | P1<br>P2          |
| 06:00 - 08:00 | P1<br>P2            | P1<br>P2              | P1<br>P2           | P1<br>P2            | P1<br>P2            | P1<br>P2            | P1<br>P2          |
| 08:00 - 09:59 | P1<br>P2            | P1<br>P2              | P1<br>P2           | P1<br>P2            | P1<br>P2            | P1 DANI C<br>P2     | P1<br>P2          |
| 10:00 - 11:59 | P1<br>P2            | P1 SERGIO LOP<br>P2   | P1<br>P2           | P1<br>P2            | P1<br>P2            | P1<br>P2            | P1<br>P2          |
| 12:00 - 13:59 | P1<br>P2            | P1<br>P2              | P1<br>P2           | P1<br>P2            | P2 SERGIO LOP<br>P1 | P1<br>P2            | P1<br>P2          |
| 14:00 - 15:59 | P1<br>P2            | P1<br>P2              | P1<br>P2           | P1<br>P2            | P1<br>P2            | P1<br>P2            | P1<br>P2          |
| 16:00 - 17:59 | P1<br>P2            | P1<br>P2              | P2 DANI C<br>P1    | P1<br>P2            | P1<br>P2            | P1<br>P2            | P1<br>P2          |
| 18:00 - 19:59 | P1<br>P2            | P1<br>P2              | P1<br>P2           | P1<br>P2            | P1<br>P2            | P1<br>P2            | P1<br>P2          |
| 20:00 - 21:59 | P1<br>P2            | P1<br>P2              | P1<br>P2           | P1<br>P2            | P1<br>P2            | P1<br>P2            | P1<br>P2          |
| 22:00 - 23:59 | P1<br>P2            | P1<br>P2              | P1<br>P2           | P1<br>P2            | P1<br>P2            | P1<br>P2            | P1<br>P2          |

Reservar Puesto  Cancelar Reserva 

**Figura 5.3-4: Panel de reservas**

- Un tabón de avisos donde los administradores del laboratorio pueden dejar mensajes de interés a los usuarios (Figura 5.3-5).



**Figura 5.3-5: Panel informativo del laboratorio**

**Acción 3:** observar el panel de reservas y comprobar la ocupación de los puestos de laboratorio. Realizar la reserva de un puesto de laboratorio en función de su disponibilidad, debe tener en cuenta que para poder completar este tutorial, a partir del apartado 5.3.4.1, es necesario que esté reservado a su nombre, en el horario que haya elegido, el puesto de laboratorio en el que va a trabajar.

- **Sala de Video**

Esta sala está preparada para reproducir videos (Figura 5.3-6). En la actualidad sólo es posible reproducir un video demostrativo sobre el funcionamiento del laboratorio eLab-3D. El video se activa cuando se pulsa la tecla "Play" del menú de visualización de videos ubicado en la parte superior derecha del visor<sup>8</sup> (Figura 5.3-7).

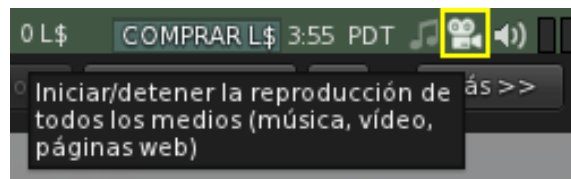
---

<sup>8</sup> Para poder visualizar correctamente el video es necesario que tenga instalado en su ordenador el reproductor QuickTime.





**Figura 5.3-6: Sala de video**



**Figura 5.3-7: Botón para iniciar videos**

- **Sala I+D:**

Sala de desarrollo para seguir ampliando funcionalidades de los puestos de laboratorio (Figura 5.3-8). El acceso a esta sala está restringido sólo para los desarrolladores.



**Figura 5.3-8: Sala de I+D**

- **Aula de Laboratorio**

Sala donde se encuentran los puestos de laboratorio que ocuparán los usuarios para realizar las diferentes actividades prácticas (Figura 5.3-9). En la versión actual de la plataforma se han preparado dos puestos de laboratorio aunque en el futuro este número es posible que aumente.



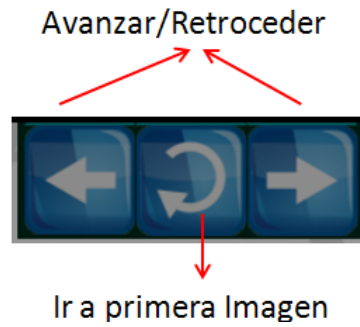
**Figura 5.3-9: Aula de laboratorio**

En esta sala se dispone también de un reproductor de diapositivas (Figura 5.3-10). Consta de una pantalla en la que se visualizan varias diapositivas explicativas para ayudar a los usuarios. El reproductor las visualiza en función de las acciones realizadas sobre los controles (Figura 5.3-11).



**Figura 5.3-10: Reproductor de diapositivas**





**Figura 5.3-11: Botones del reproductor**

**Acción 4:** acceder al Aula de laboratorio. Si la puerta está cerrada podrá abrirla tocándola mediante un clic de ratón. Ya en el aula utilizar los botones del proyector para ver alguna diapositiva de la presentación.

### 5.3.4 Puesto de laboratorio

Los objetos asociados al puesto de laboratorio se muestran en la Figura 5.3-12 y se describen brevemente a continuación:



**Figura 5.3-12: Puesto de laboratorio**

- **Mesa**

Sólo sirve de soporte para colocar los instrumentos y las cajas de componentes. Posee una serie de cajones donde se almacenan los cables que se utilizan para interconectar los instrumentos con las placas de prueba.

- **Monitor informativo**

Muestra datos referentes al puesto y al estado del servidor (Figura 5.3-13). Si el puesto está operativo al lado de la etiqueta “Estado Servidor” aparece la palabra “ON”, si, por el contrario, el puesto no se encuentra operativo, al lado de la etiqueta “Estado Servidor” aparece la palabra “OFF”. Si el puesto ha sido asignado a algún usuario aparece en la pantalla del monitor el nombre del usuario.



**Figura 5.3-13: Monitor informativo**

- **Caja de componentes**

Se almacenan en ellas los diferentes componentes electrónicos que se utilizarán para configurar las diferentes placas de pruebas.

- **Armario**

Lugar donde se almacenan, en diferentes cajones, las placas de pruebas que se utilizarán en las diferentes actividades prácticas.

- **Instrumentos**

Se dispone de los típicos equipos de excitación (fuente de alimentación y generador de funciones) y medida (multímetro y osciloscopio) habituales en cualquier laboratorio de electrónica.

Los equipos incluyen la mayoría de las funcionalidades de los instrumentos reales, estando deshabilitadas algunas de ellas por no ser necesarias o no estar implementadas cuando se utiliza el equipo de forma remota. Para ayudar a conocer cuáles son las opciones permitidas en los equipos, éstas se han destacado con un color gris claro, mientras que las funcionalidades no implementadas se reflejan en un color gris oscuro. A modo de ejemplo, en la Figura 5.3-14 se muestra el panel frontal del generador de funciones del laboratorio donde se pueden observar qué botones tienen funcionalidad y cuáles no.



**Figura 5.3-14: Ejemplo de instrumento**

Una vez descritos los objetos del puesto de laboratorio se va a proceder a explicar de forma detallada cómo poder interactuar con todos ellos.

#### **5.3.4.1    *Iniciar sesión (log in)***

Para poder acceder al puesto de laboratorio es necesario solicitar la asignación del puesto al servidor. Para iniciar la sesión sólo hay que tocar la silla del puesto que se ha reservado.

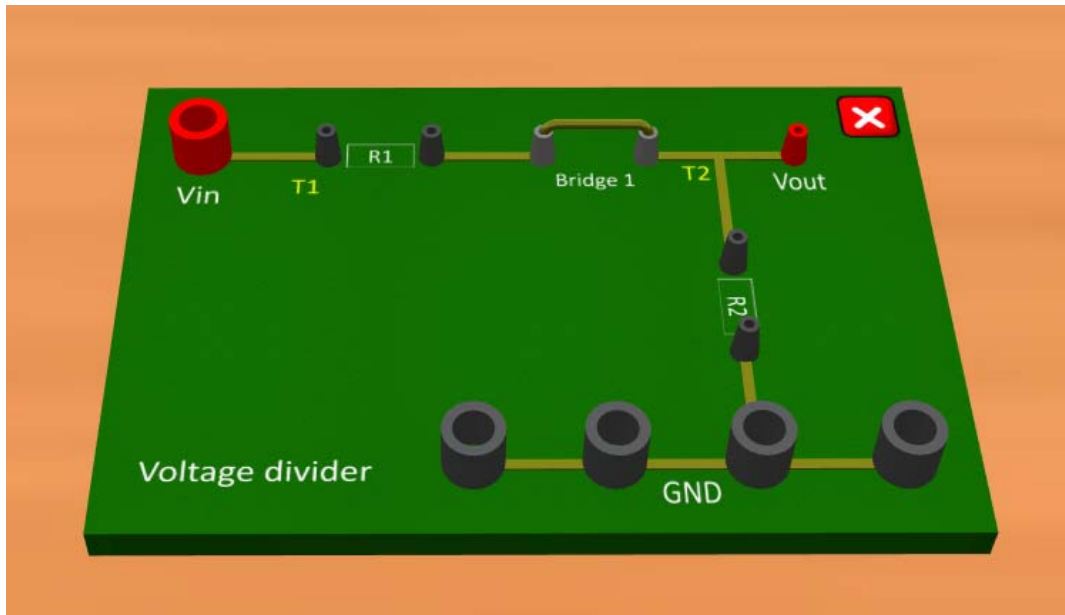
Al tocar la silla el puesto quedará asignado para dicho usuario, siempre que éste figure en la base de datos del servidor del laboratorio y sea el usuario que ha reservado el puesto. A partir de este momento ya se podrá interactuar con todos los objetos del puesto.

**Acción 5:** con el avatar en el aula de laboratorio tocar la silla del puesto de laboratorio reservado para que el servidor le asigne el puesto y se inicie la sesión.

#### **5.3.4.2    *Selección de la placa de prueba***

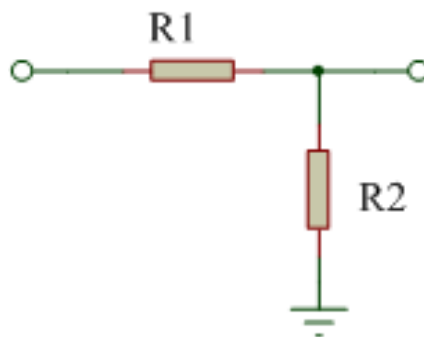
Para comenzar a realizar alguna actividad práctica es necesario elegir una determinada placa de pruebas. En el armario del puesto, en el interior de los cajones existentes, se encuentran las placas de pruebas que pueden ser utilizadas.

**Acción 6:** abrir el cajón con la etiqueta “Divisor de Tensión” y tocar la placa que hay en el interior. Dicha placa se colocará automáticamente en la mesa del puesto (Figura 5.3-15).



**Figura 5.3-15: Placa del divisor de tensión**

El circuito que se va a probar es un divisor de tensión resistivo cuyo esquema se muestra en la Figura 5.3-16.



**Figura 5.3-16: Esquemático del divisor de tensión**

**Acción 7:** puede retirar la placa del puesto simplemente pulsando sobre el aspa ubicado en la parte superior derecha de la placa. Retire la placa y vuelva a situarla de nuevo sobre la mesa para poder continuar con el tutorial.

### 5.3.4.3 *Configuración de la placa*

La placa inicialmente no incluye los componentes ya que es el usuario quién debe colocarlos en función de las especificaciones que deba cumplir el circuito.

En la parte izquierda de la mesa se encuentran los cajones que contienen las cajas de componentes que se podrán utilizar según la placa seleccionada (Figura 5.3-17). Los cajones tienen el nombre de los componentes que contienen.



**Figura 5.3-17: Cajones de componentes**

Para insertar los componentes en la placa es necesario sacar la caja del cajón de componentes adecuada. Al tocar la caja, ésta se sitúa automáticamente en la mesa, cerca de la placa. A continuación se deben seguir los siguientes pasos:

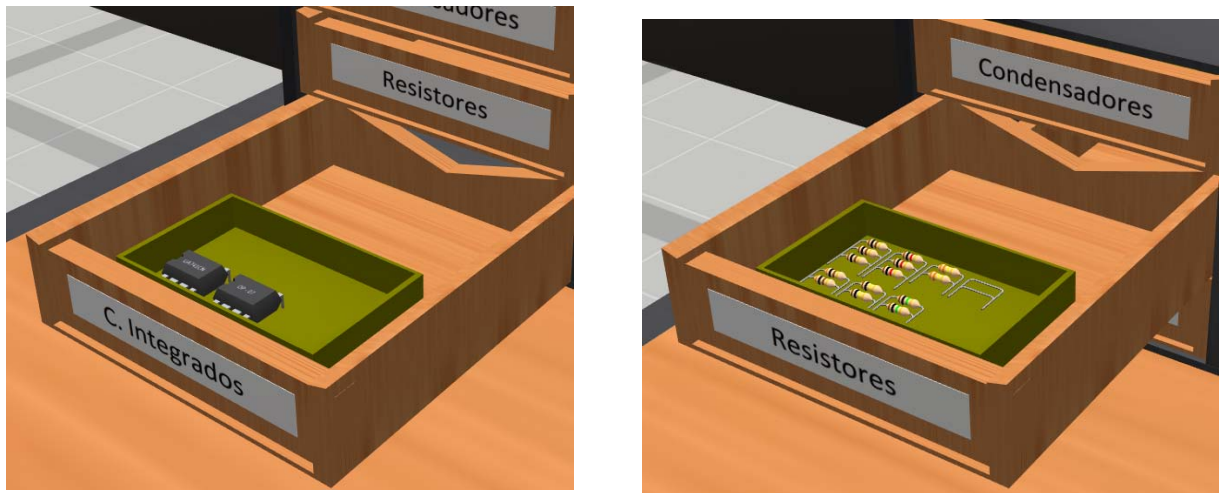
- Elegir, del interior de la caja, el componente que se desea colocar tocando sobre él, se pondrá brillante<sup>9</sup> para indicar cuál se ha seleccionado.
- Tocar en el hueco de la placa donde se desea colocar el componente. Se insertará en dicho hueco siempre que sea válido para dicho componente. Desde que selecciona el componente hasta que lo conecta en la placa dispone de un tiempo de cinco segundos. Transcurrido ese tiempo, si el componente no se ha conectado en la placa, habrá que seleccionarlo de nuevo.

---

<sup>9</sup> Este efecto visual puede ser más o menos pronunciado en función de las preferencias programadas en el visor y de las características de la tarjeta gráfica de su ordenador.

En la mesa solo puede haber una caja de componentes a la vez. Para devolver la caja de componentes a su cajón correspondiente sólo es necesario tocarla.

En la Figura 5.3-18 se muestran las cajas de componentes que contienen circuitos integrados y resistores.

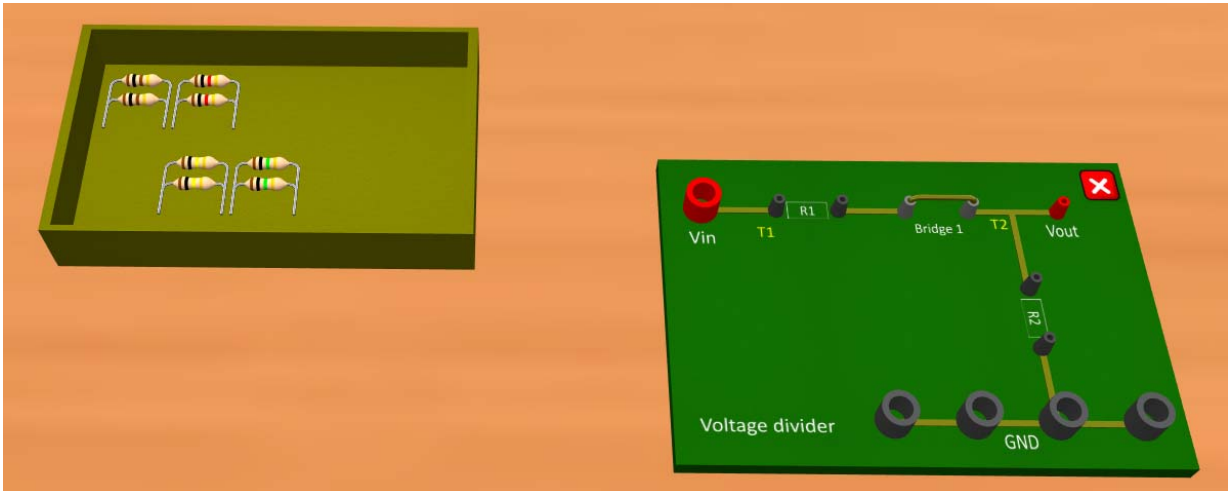


**Figura 5.3-18: Cajones de componentes abiertos**

Los puentes que existan en la placa (Bridge 1 en Figura 5.3-15), por defecto aparecerán siempre conectados. Sólo cuando se desee realizar medidas de corriente será necesario retirarlos.

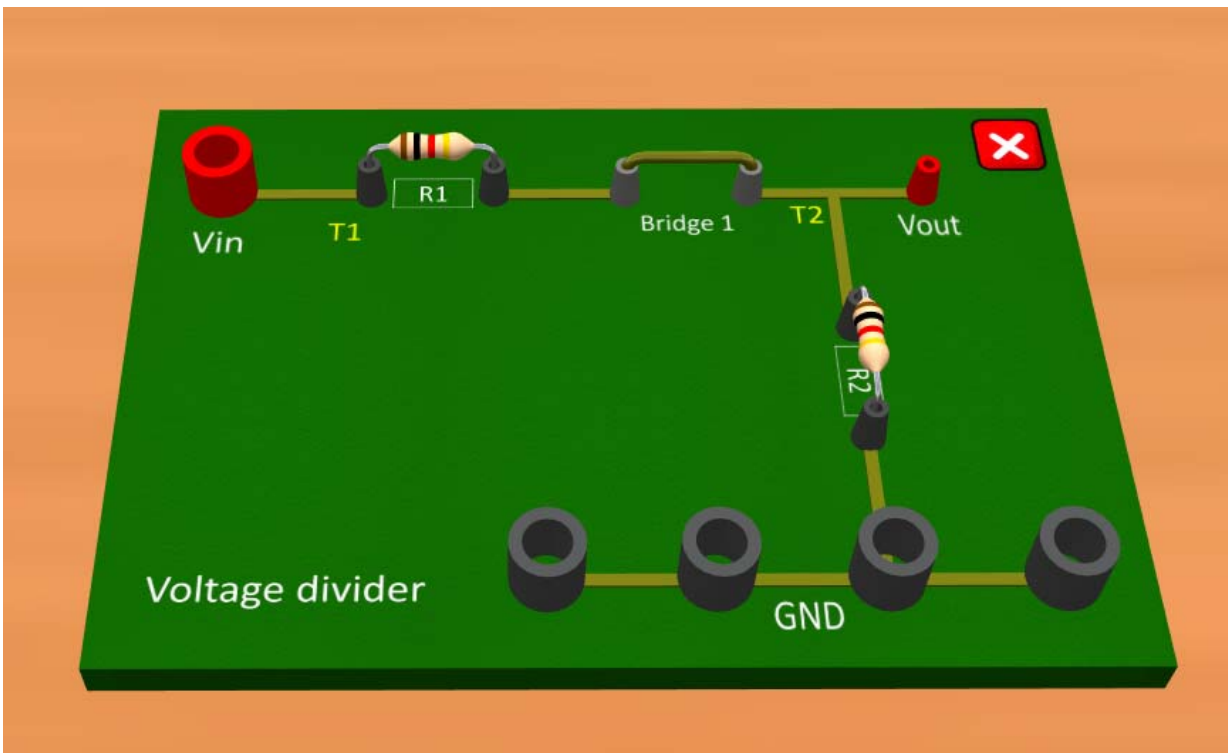
**Acción 8:** elegir el cajón de resistores. Tocar la caja que contiene los resistores para que se sitúe en la mesa. Manipular la cámara para obtener una perspectiva de la caja de resistores y la placa Divisor de tensión como se muestra en la Figura 5.3-19.





**Figura 5.3-19: Resistores y placa divisor de tensión**

Seleccionar un resistor de  $1\text{k}\Omega$  (código de colores: marrón, negro, rojo) y situarlo en el espacio de la placa donde está la etiqueta  $R1^{10}$ . Realizar la misma operación para insertar otro resistor de  $1\text{k}\Omega$  en  $R2$ . La placa debe quedar configurada como se muestra en la Figura 5.3-20.



**Figura 5.3-20: Divisor de tensión con resistores**

<sup>10</sup> Esta acción no es instantánea puede que tarde algunos segundos.



#### 5.3.4.4 Selección de los cables

En el interior de los cajones situados en la parte inferior de la mesa se encuentran los cables que permiten la conexión entre los instrumentos y las placas de prueba (Figura 5.3-21). Cada cajón está identificado con una etiqueta que indica el tipo de cables que contiene. Para colocarlos en la mesa solo es necesario tocar el cable correspondiente.



**Figura 5.3-21: Cajón de cables**

**Acción 9:** sacar los cables de la fuente de +6V, generador de funciones, multímetro y osciloscopio.

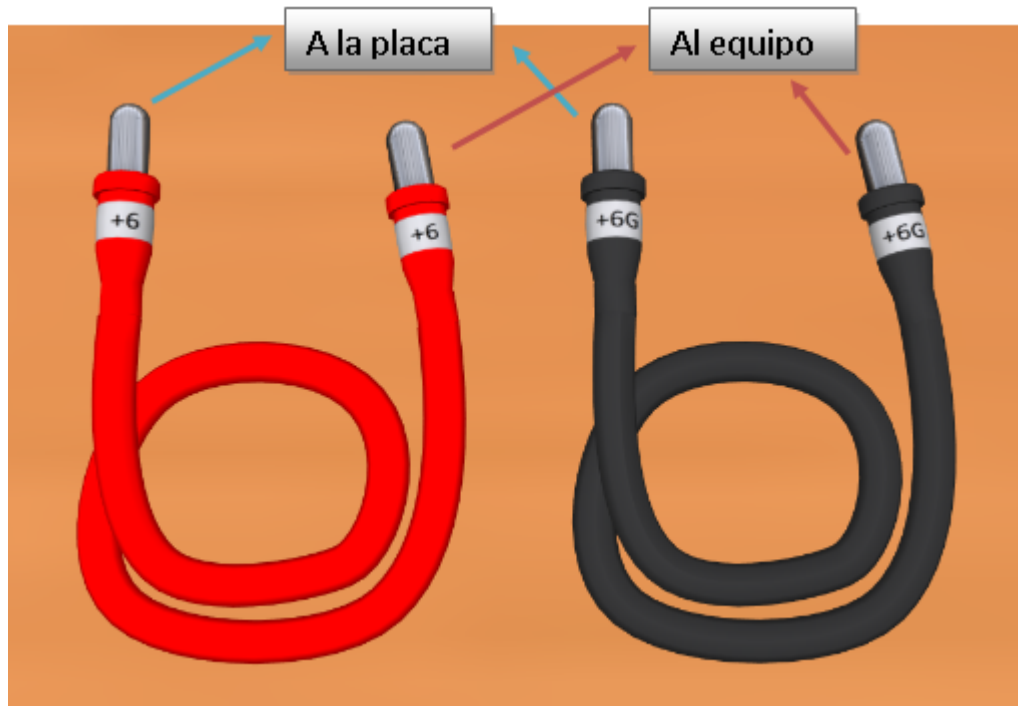
#### 5.3.4.5 Conexión y desconexión de los cables

Una vez estén los cables en la mesa, la forma de realizar una conexión consiste en tocar **siempre primero** el terminal del cable correspondiente y **después** el conector del equipo o el punto de conexión de la placa, teniendo en cuenta que hay 5 segundos para realizar la conexión. El brillo<sup>11</sup> del terminal indica cuál está seleccionado.

---

<sup>11</sup> Este efecto visual puede ser más o menos pronunciado en función de las preferencias programadas en el visor y de las características de la tarjeta gráfica de su ordenador.

En los cables de la fuente de alimentación (Figura 5.3-22) y multímetro los terminales de la izquierda de cada cable sólo se pueden conectar en la placa de pruebas y los de la derecha sólo en los equipos.



**Figura 5.3-22: Cables fuente de alimentación**

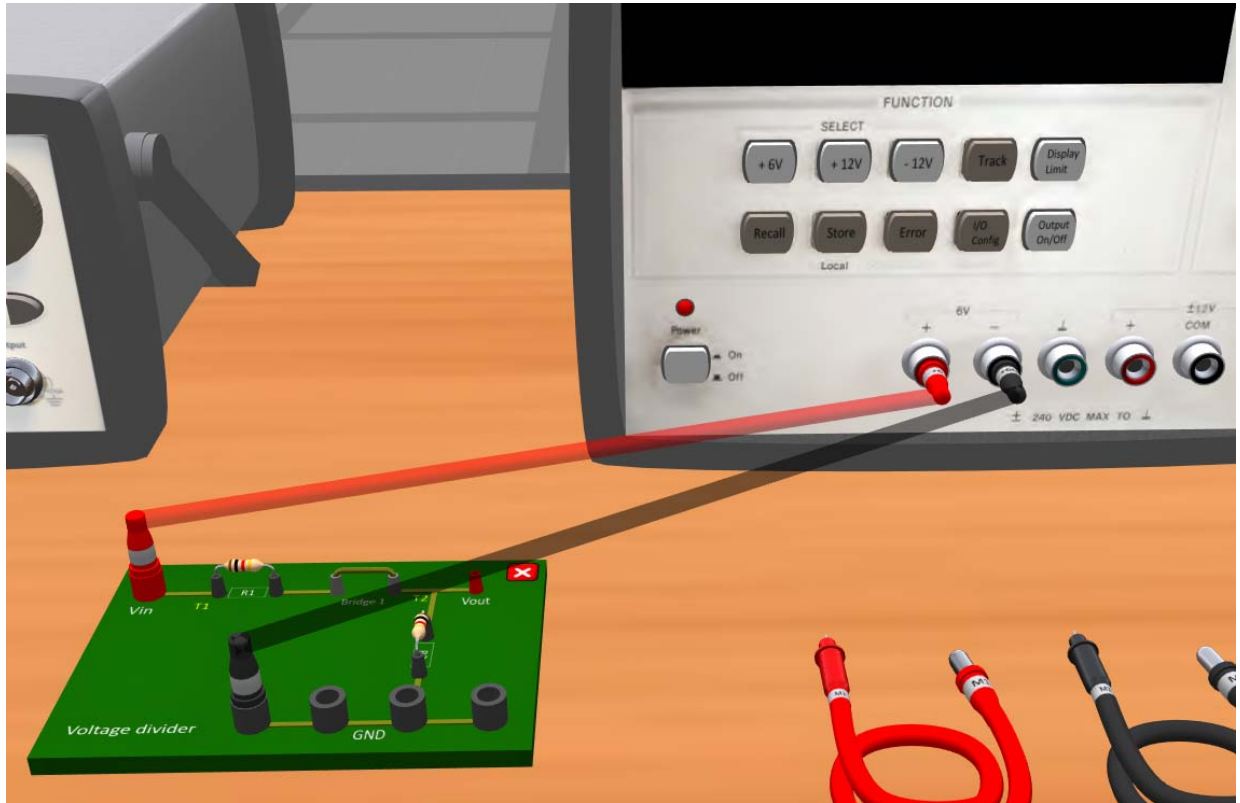
En los cables del generador y el osciloscopio los terminales negros sólo se pueden conectar a la masa del circuito.

Para desconectar los cables es necesario tocar los dos terminales del mismo cable de forma consecutiva, de esta forma el cable volverá a colocarse en la mesa.

**Acción 10:** conectar la salida de +6V de la fuente de alimentación al Divisor de Tensión como se muestra en la Figura 5.3-23, utilizando el cable rojo +6 y el cable negro +6G.

La banana derecha del cable rojo +6 se conectará al terminal (+) de la salida +6V de la fuente de alimentación, y la banana izquierda de dicho cable en la entrada del circuito Vin.

La banana derecha del cable negro +6G se conectará al terminal (-) de la salida +6V de la fuente de alimentación, y la banana izquierda de dicho cable en la masa (GND) del circuito.



**Figura 5.3-23: Conexión de los cables de la fuente de alimentación**

**Acción 11:** desconecte los cables de la salida de +6V de la fuente de alimentación para que se queden en la mesa.

#### **5.3.4.6 Configuración de la Fuente Alimentación**

La fuente de alimentación funciona como fuente de voltaje constante siempre que la corriente que demande el circuito en cada momento no supere el límite de corriente máxima que se haya establecido previamente.

Consta de tres salidas independientes, +6V, +12V y -12V. Las fuentes de  $\pm 12V$  se utilizarán para proporcionar la alimentación de los circuitos y la fuente de +6V se

podrá utilizar como entrada de excitación continua. En la Figura 5.3-24 se muestra el panel frontal de la fuente de alimentación.



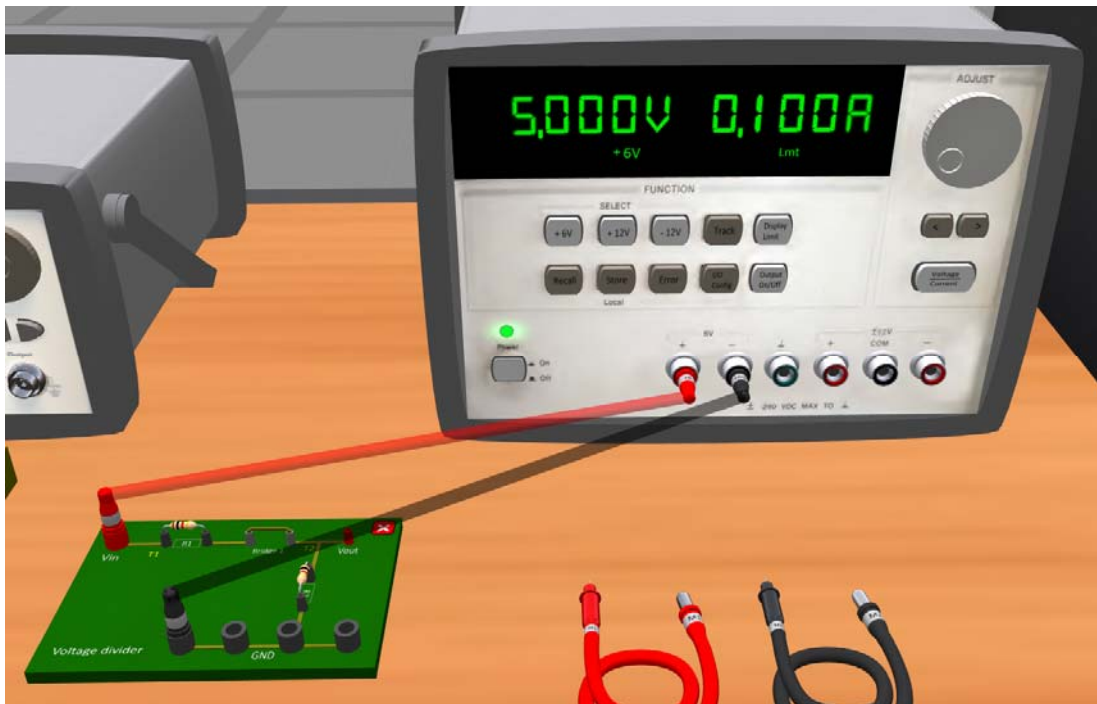
Figura 5.3-24: Fuente de alimentación

***Los valores de tensión/corriente de las fuentes de +12V y -12 V que se utilizan para la alimentación de diversas placas son fijos y no se pueden modificar.***

Para configurar el voltaje o el límite de corriente de la salida de +6 V hay que utilizar el botón “Display Limit”. La unidad que parpadea (V ó A), indica si se está modificando la tensión o la corriente, pudiendo cambiar entre ellos mediante el botón “Voltage/Current”. Para asignar un valor de voltaje o corriente hay que pulsar el botón giratorio de la parte superior del equipo e introducir el valor desde el menú que aparece en la esquina superior derecha. En modo voltaje el valor debe estar comprendido entre 0 y 6 V y en modo corriente entre 0 y 5 A.

**Acción 12:** encender la fuente de alimentación y configurar la salida de +6V para que proporcione a su salida 5 V. Limitar la corriente máxima a 100 mA.

**Acción 13:** conectar la salida de +6V de la fuente de alimentación, mediante los cables adecuados, a la entrada del Divisor de Tensión como se muestra en la Figura 5.3-25.



**Figura 5.3-25: Conexión de los cables de la fuente de alimentación**

Una vez configurados los valores deseados y conectados los cables en la placa, con el botón “Output On/Off”, se puede habilitar o deshabilitar la salida de la fuente de alimentación. Al habilitar la salida en el display de la fuente aparece la etiqueta “ON” (Figura 5.3-26).

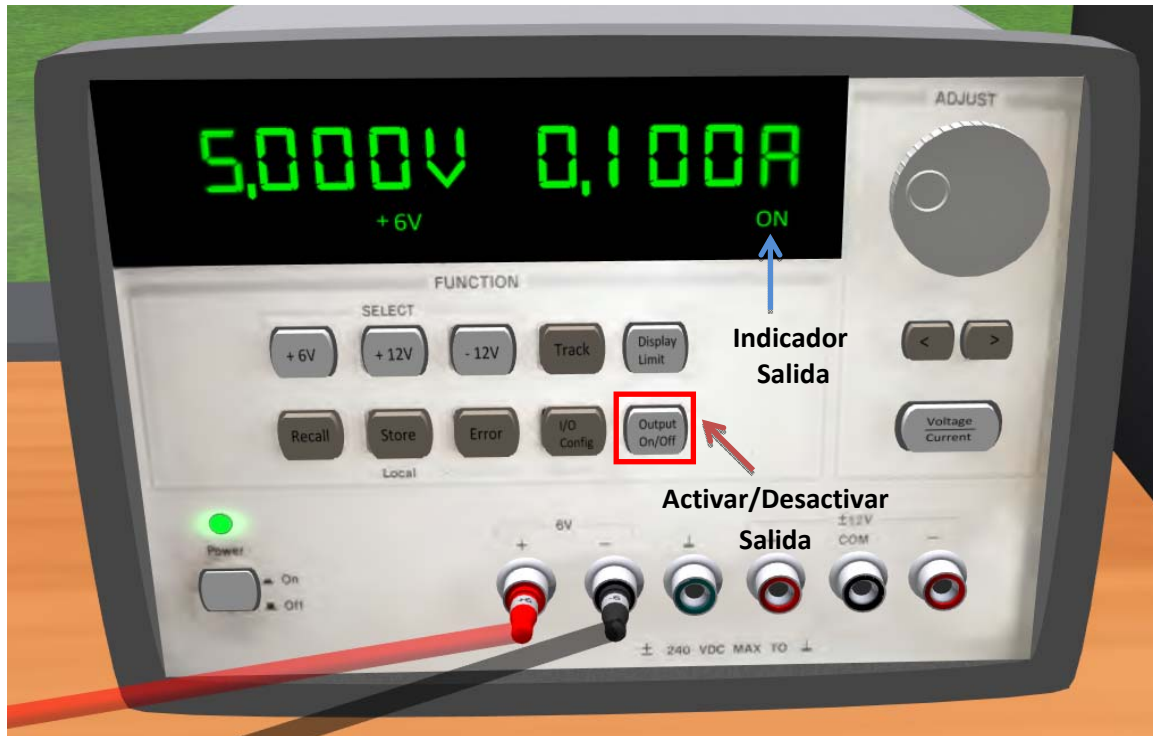
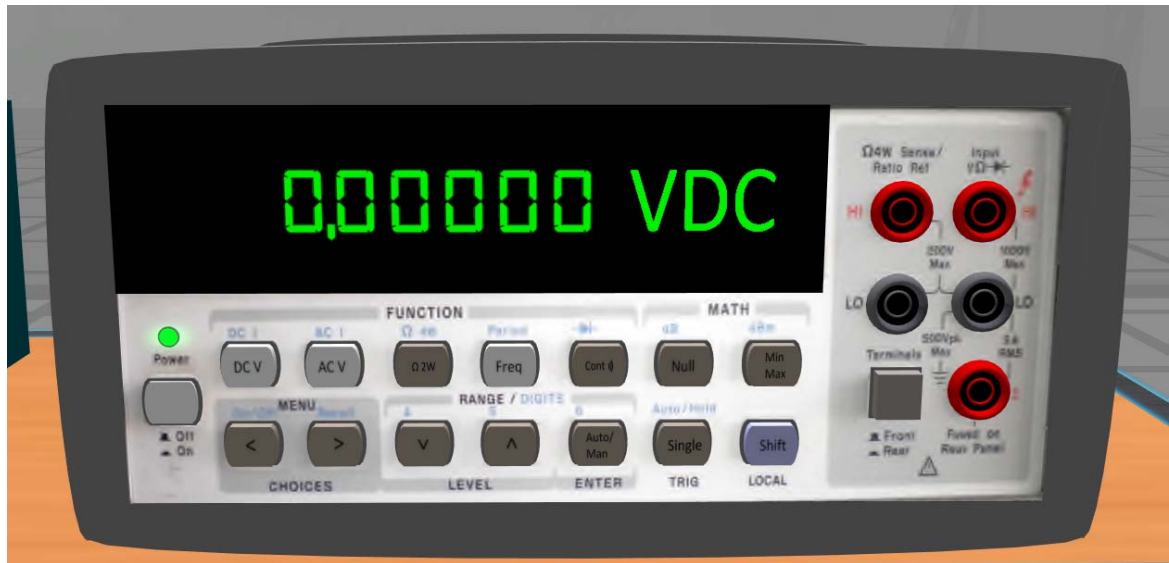


Figura 5.3-26: Habilidad de la fuente de alimentación

**Acción 14:** pulsar el botón Output On/Off para habilitar la salida de la fuente de alimentación.

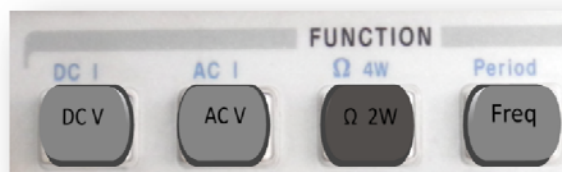
#### 5.3.4.7 Medidas con el Multímetro

Permite la realización de diferentes tipos de medida. En concreto se puede elegir entre medidas de tensión y corriente en AC y DC, frecuencia y periodo. Para realizar cualquiera de las medidas anteriores primero habrá que seleccionar el tipo de medida mediante los botones del equipo y después colocar los cables en los conectores adecuados. En la Figura 5.3-27 se muestra el panel frontal del multímetro.



**Figura 5.3-27: Multímetro**

Las medidas que se pueden realizar con el multímetro, al pulsar los botones del panel frontal del instrumento que se muestran en las figuras Figura 5.3-28: Tipos de medida del multímetro y Figura 5.3-29, son las siguientes:



**Figura 5.3-28: Tipos de medida del multímetro**



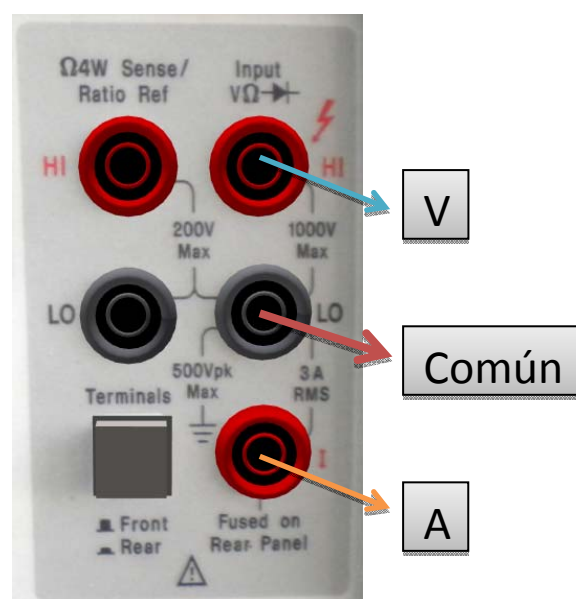
**Figura 5.3-29: Botón Shift del multímetro**



- Voltaje continuo: pulsar el botón con la etiqueta DC V
- Voltaje alterno: pulsar el botón con la etiqueta AC V
- Frecuencia: pulsar el botón con la etiqueta Freq
- Corriente continua: pulsar en primer lugar el botón con la etiqueta Shift y después el botón con la etiqueta DC V
- Corriente alterna: pulsar en primer lugar el botón con la etiqueta Shift y después el botón con la etiqueta AC V
- Periodo: pulsar en primer lugar el botón con la etiqueta Shift y después el botón con la etiqueta Freq

El multímetro posee los conectores que se muestran en la Figura 5.3-30. Los terminales de los cables se deben conectar correctamente de acuerdo al tipo de medida que desee realizarse:

- Conexión de terminales de los cables para medidas de tensión: utilizar el conector común (negro) y el conector “V” (rojo) que se encuentra encima del común.
- Conexión de terminales de los cables para medidas de corriente: utilizar el conector común (negro) y el conector “A” (rojo) que se encuentra debajo del común.



**Figura 5.3-30: Conectores del multímetro**



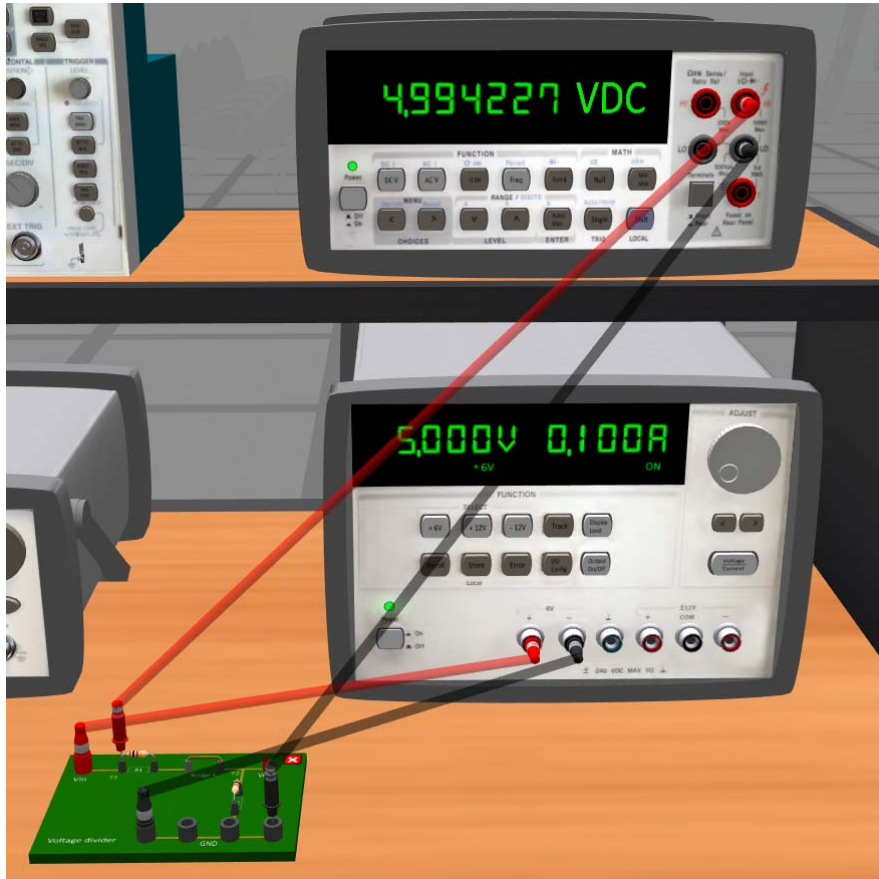
Es importante tener en cuenta que a la hora de pasar de medir tensión a corriente o viceversa es necesario quitar los cables y a continuación configurar el tipo de medida que se desea realizar. Si se cambia de tipo de medida con los cables conectados, éstos se retirarán a la mesa automáticamente para proteger el equipo de medida y se deberá realizar la conexión de los cables de nuevo.

**Acción 15:** determinar en la placa Divisor de Tensión el valor de tensión que debe existir en  $V_{in}$  teniendo en cuenta el montaje realizado hasta el momento mostrado en la Figura 5.3-25. Anotarlo a continuación:

***Voltaje en  $V_{in}$ :***

**Acción 16:** encender el multímetro pulsando el botón “Power”, por defecto aparecerá configurado para medir tensión en continua, no cambie el tipo de medida.

**Acción 17:** sacar del cajón correspondiente los cables del multímetro. Conectarlos para medir, en la placa Divisor de Tensión, el valor del voltaje en  $V_{in}$  (situar la punta de prueba del cable en la patilla izquierda de **R1**, punto de test T1), como se indica en la Figura 5.3-31.



**Figura 5.3-31: Conexión del multímetro a la entrada de la placa**

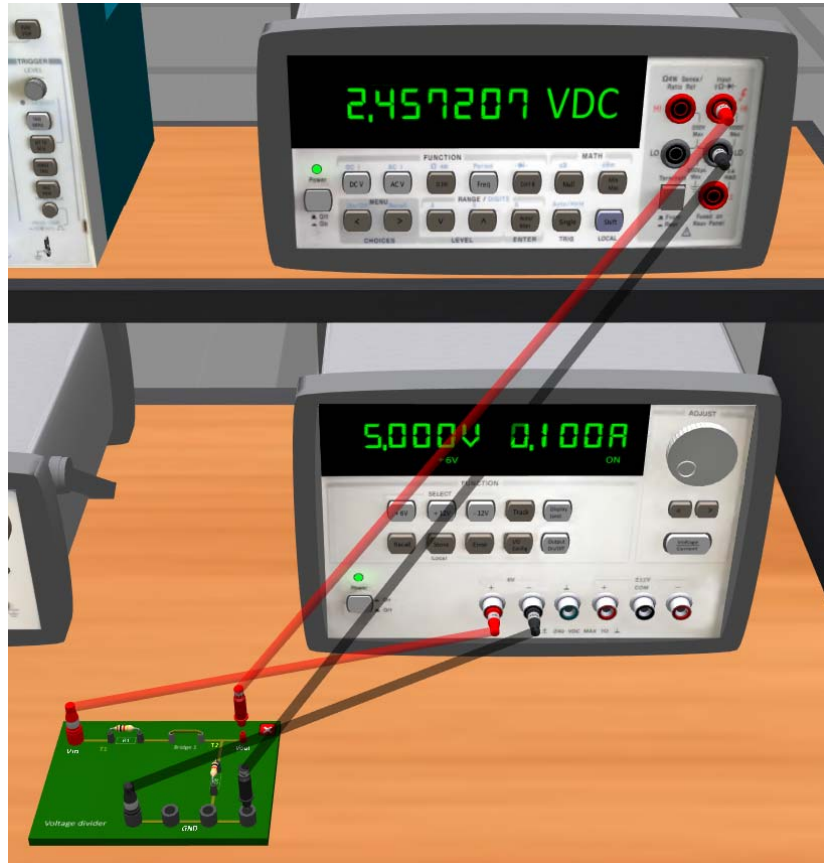
**Acción 18:** anotar el voltaje medido por el multímetro comprobando que se corresponde con el valor previsto.

#### ***Voltaje medido en $V_{in}$ :***

**Acción 19:** determinar, de forma teórica, en la placa Divisor de Tensión el valor de tensión que debe existir en  $V_{out}$ . Anotarlo a continuación:

#### ***Voltaje en $V_{out}$ :***

**Acción 20:** conectar los cables del multímetro para medir el valor del voltaje en  $V_{out}$  como se indica en la Figura 5.3-32. Para hacer esta medida simplemente toque la punta de prueba del cable rojo situada en  $V_{in}$  y luego toque en el conector de la placa  $V_{out}$  (punto de test T2).



**Figura 5.3-32: Conexión del multímetro a la salida de la placa**

**Acción 21:** anotar el voltaje medido por el multímetro comprobando que se corresponde con el valor previsto.

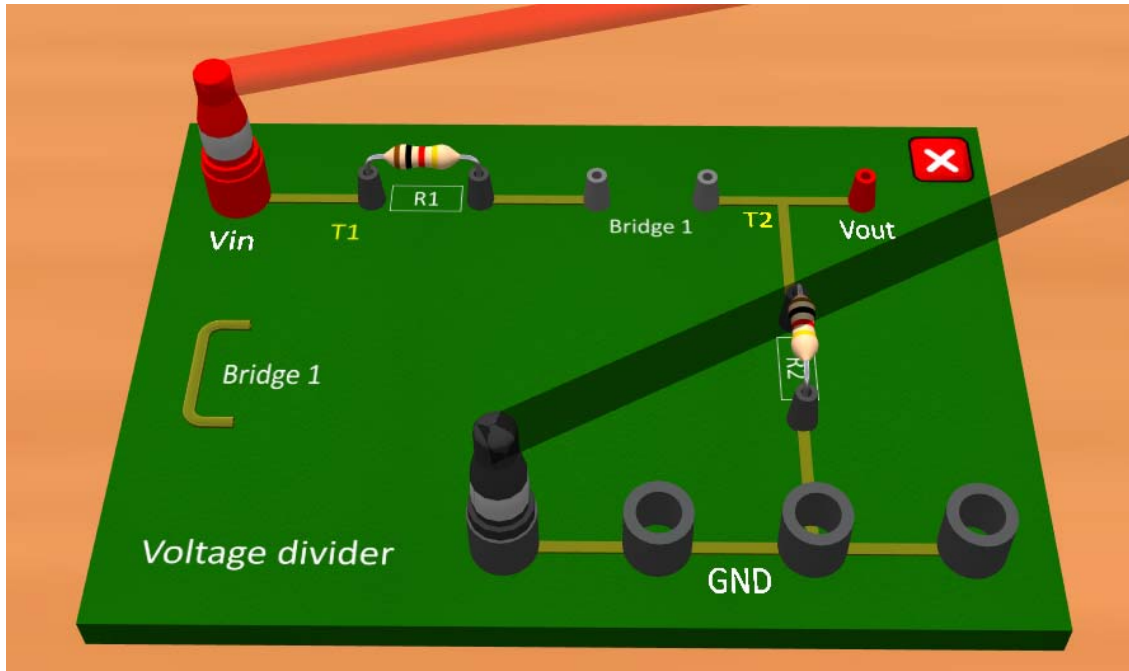
***Voltaje medido en Vout:***

**Acción 22:** calcular, aplicando la Ley de Ohm, la corriente que circula por los resistores R1 y R2. Anotar el resultado a continuación:

***Valor calculado de IR1:***

**Acción 23:** realizar la medida de la corriente que circula por el divisor de tensión. En primer lugar desconectar los cables del multímetro que estaban conectados para medir voltaje. Recordar que para desconectar los cables y que se sitúen en la mesa es necesario tocar los dos terminales del mismo cable de forma consecutiva.

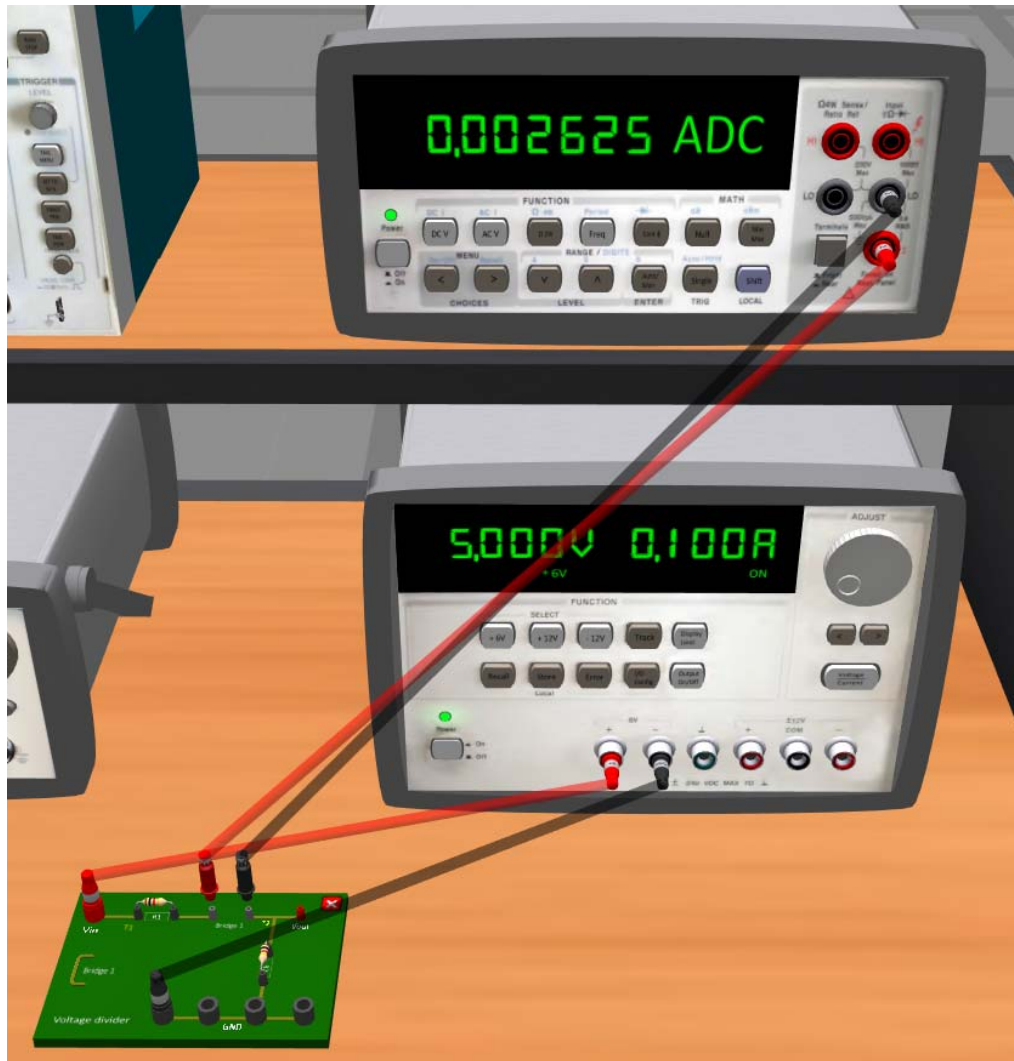
**Acción 24:** retirar el puente (Bridge 1) que hay en el circuito, haciendo clic sobre el mismo, para poder posteriormente conectar los cables del multímetro y medir corriente. Observar la Figura 5.3-33 donde se muestra que el puente se ha retirado del circuito.



**Figura 5.3-33:** Placa con el puente 1 quitado

**Acción 25:** configurar el multímetro para que mida corriente continua pulsando el botón SHIFT y después el botón DC V.

**Acción 26:** conectar los cables para medir la corriente que circula por el divisor de tensión como se indica en la Figura 5.3-34.



**Figura 5.3-34: Medida de corriente con el multímetro**

**Acción 27:** anotar la corriente medida por el multímetro comprobando que se corresponde con el valor previsto.

***Voltaje medido de IR1:***

**Acción 28:** desconectar los cables y apagar la fuente de alimentación y el multímetro.

### 5.3.4.8 Configuración del Generador de Funciones

El generador de funciones se utilizará para obtener las señales periódicas que se aplicarán a los diferentes circuitos. Las señales que puede generar son de cuatro tipos: sinusoidal, cuadrada, triangular y diente de sierra. Los parámetros que se pueden configurar de cada forma de onda son: frecuencia, amplitud y offset (valor del voltaje continuo). En la Figura 5.3-35 se muestra el panel frontal del generador de funciones.



**Figura 5.3-35: Generador de señales**

En el panel frontal se pueden identificar los controles que permiten realizar las siguientes funciones:

- Selección de la forma de onda a generar. En la Figura 5.3-36: Formas de onda e muestran los botones que hay que pulsar para poder generar una onda sinusoidal, cuadrada, triangular y diente de sierra.



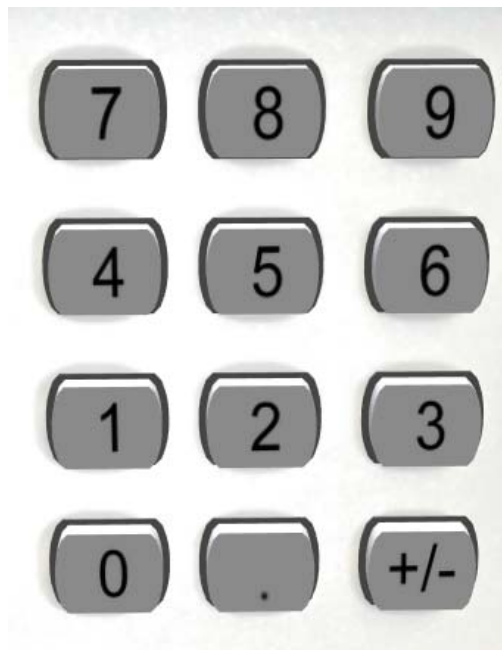
**Figura 5.3-36: Formas de onda**

- Selección de los parámetros de configuración de la forma de onda. En la Figura 5.3-37: Parámetro del generador se muestran los botones que hay que pulsar para poder configurar la frecuencia, amplitud, offset y ciclo de trabajo (%Duty) de la señal.



**Figura 5.3-37: Parámetro del generador**

- Introducción del valor numérico del parámetro elegido. En la Figura 5.3-38 se muestra el teclado numérico del instrumento que permite introducir el valor del parámetro de configuración seleccionado.



**Figura 5.3-38: Teclado numérico del generador de señales**

- Selección de las unidades del parámetro elegido. En la Figura 5.3-39 se muestran los botones que hay que pulsar para elegir la unidad que se muestra en la parte inferior del display del instrumento.





**Figura 5.3-39: Botones para la elección de las unidades del generador**

Los límites de los diferentes parámetros de configuración son los siguientes:

- Frecuencia: entre 1 y 15Mhz para señales sinusoidales y cuadradas. Entre 1 y 100kHz para señales triangulares y diente de sierra.
- Amplitud: entre -10V y +10 V (con offset 0V)
- Offset: menor que dos veces la amplitud programada.
- Máxima salida:  $\pm 10V$
- Activación de la salida del generador. En la Figura 5.3-40 se muestra el botón que hay que pulsar para activar la salida programada en el generador. En el display del instrumento aparecerá la etiqueta "ON" para indicar que la salida está activada.





**Figura 5.3-40: Habilitación de la salida del generador**

Si se introduce algún valor que esté fuera del rango permitido se notifica mediante un mensaje de error.

Para conectar la salida del generador de funciones a los diferentes circuitos se utilizará un cable coaxial (Figura 5.3-41) que en uno de sus extremos está terminado en un conector BNC, y en el otro extremo en dos conductores finalizados en dos bananas. El conductor activo, de color rojo, se conectará en la entrada del circuito y el conductor de masa, de color negro, se conectará a la masa del circuito.



**Figura 5.3-41: Cable del generador de funciones**

**Acción 29:** encender el generador de funciones y programar una señal sinusoidal de amplitud 1V (2Vpp), con un nivel de continua de 1V y frecuencia 2 kHz. Sacar del cajón de la mesa el cable del generador de funciones.

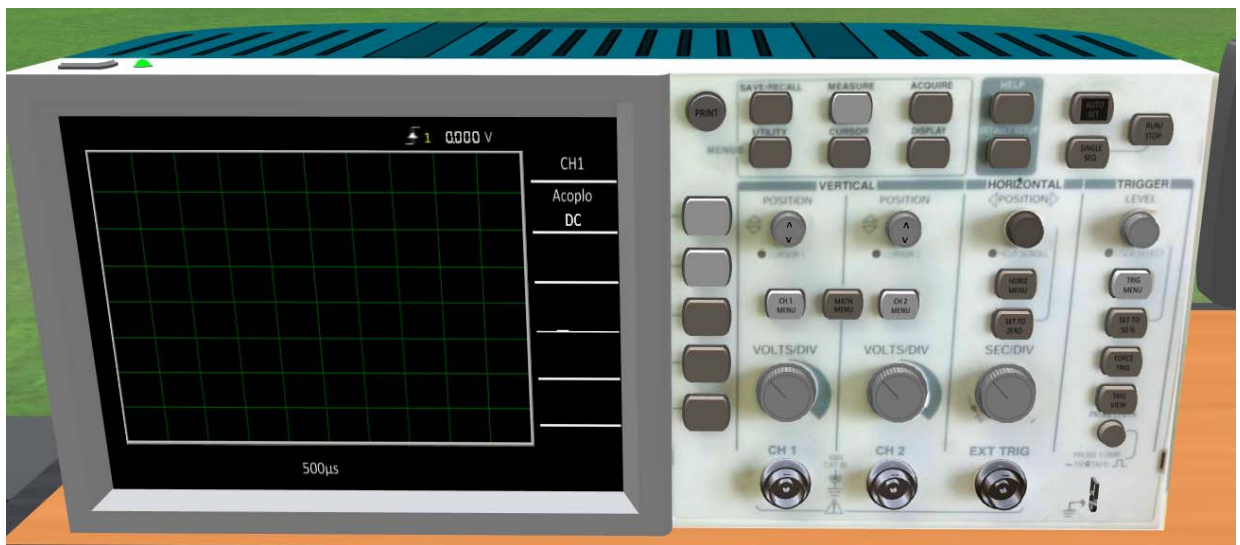
**Acción 30:** conectar mediante el cable la salida del generador de funciones a la entrada del divisor de tensión ( $V_{in}$ ) y activar la salida. La conexión realizada se muestra en la Figura 5.3-42.



**Figura 5.3-42: Conexión del generador como excitación de la placa**

### 5.3.4.9 Configuración del Osciloscopio

El osciloscopio se utilizará para visualizar las formas de onda de las señales existentes en los diferentes puntos de test de las placas de prueba. Consta de dos canales independientes y, además de la visualización de señales, permite la realización de medidas como valor pico a pico, amplitud, máximo, mínimo, valor medio, valor eficaz, frecuencia y periodo. En la Figura 5.3-43 se muestra el panel frontal del osciloscopio del laboratorio.

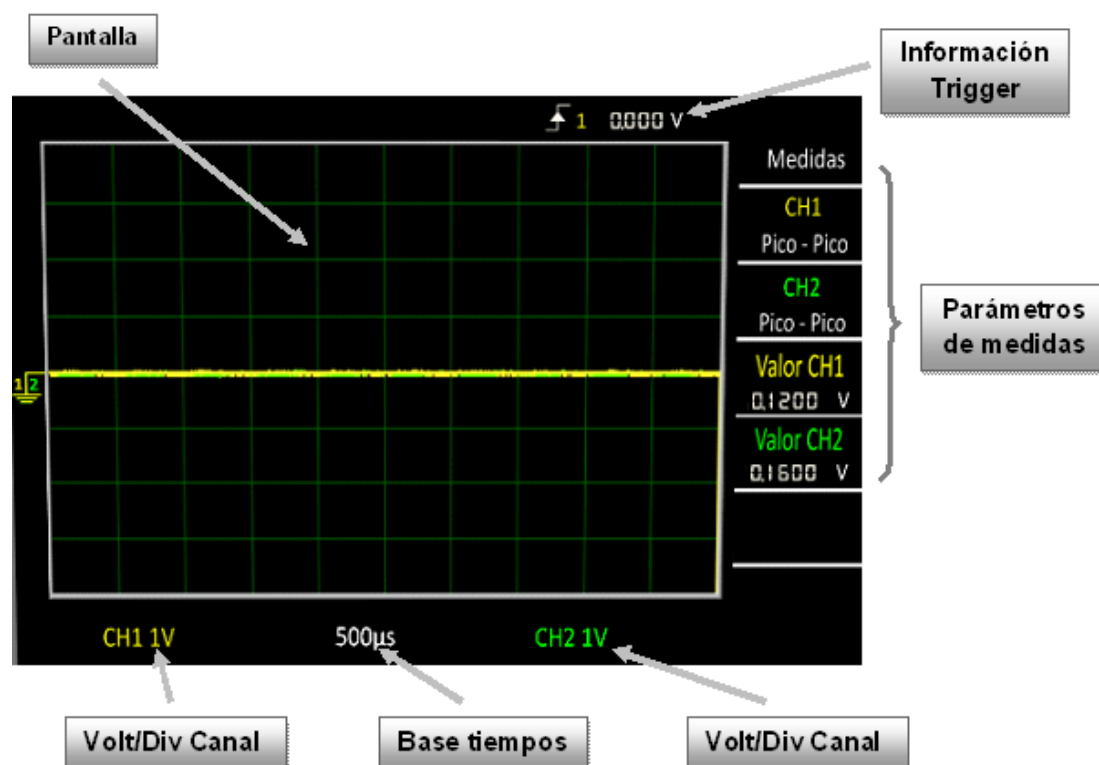


**Figura 5.3-43: Osciloscopio**

A continuación se describen brevemente los elementos clave relacionados con el osciloscopio:

#### Pantalla

En ella se representan las formas de onda de las señales que se desean visualizar y se ofrece información sobre la configuración de la base de tiempos y la resolución vertical (ver Figura 5.3-44). A su derecha existen diferentes menús interactivos cuya funcionalidad se comentará en un punto posterior.



**Figura 5.3-44: Pantalla del osciloscopio**

### Base de tiempos

Para seleccionar la base de tiempos del osciloscopio es necesario tocar el mando giratorio que se muestra en la Figura 5.3-45 y elegir el valor deseado en el menú que aparece en la parte superior derecha de la ventana del visor.

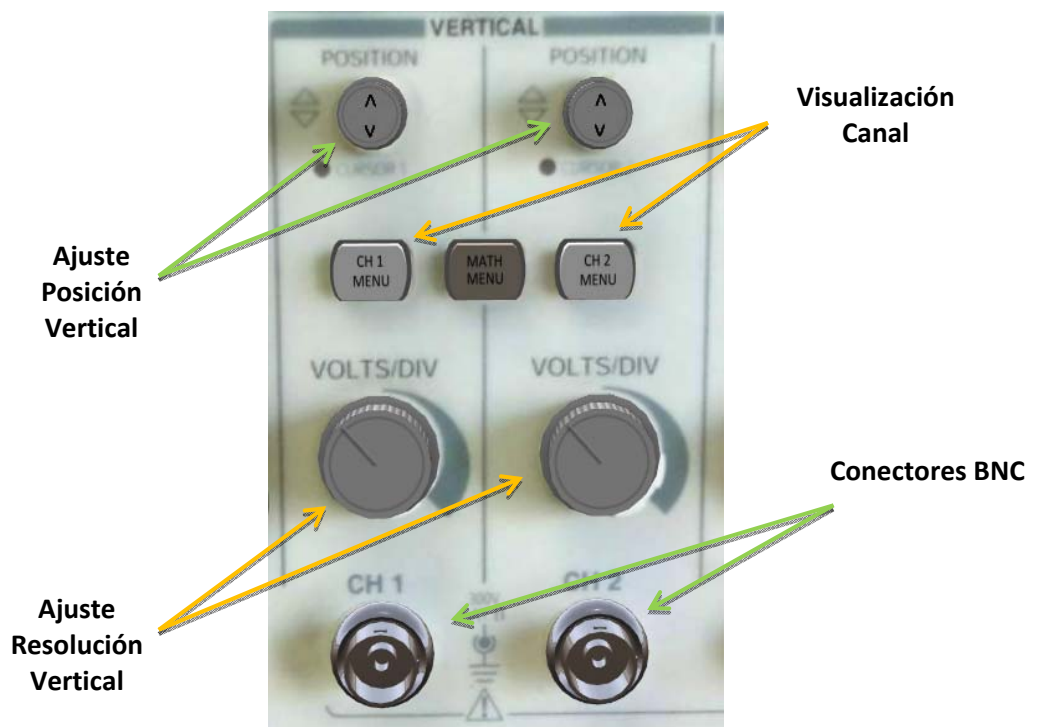


**Figura 5.3-45: Rueda para el ajuste de la base de tiempos**

### Configuración de los canales

Las opciones de configuración disponibles que afectan a cada canal son las siguientes (Figura 5.3-46):

- Botones “CH1 MENU” y “CH2 MENU”. Al pulsarlos se controlará la visualización de la traza del canal correspondiente.
- Mando giratorio “POSITION”. Al tocar las flechas subirá o bajara el nivel de referencia de cada canal hasta un máximo de 3 posiciones.
- Mando giratorio “VOLTS/DIV”. Al tocarlas aparecerá un menú que permitirá el ajuste de la resolución vertical de cada canal.



**Figura 5.3-46: Botones de configuración de los canales del osciloscopio**

### Control del trigger

Las opciones de configuración del trigger disponibles son las siguientes (Figura 5.3-47):

- Botón “TRIG MENU”. Al pulsarlo se despliega un menú en el que podemos elegir el canal y flanco de disparo.
- Mando giratorio “LEVEL”. Al tocarlo aparecerá un menú que permitirá el ajuste del valor de la tensión de trigger.



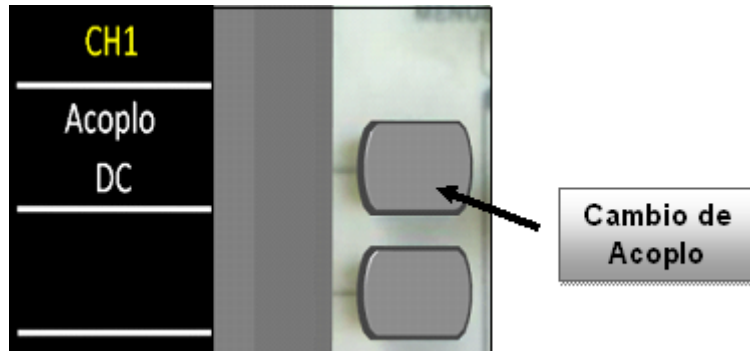
**Figura 5.3-47: Botones de configuración del trigger**

### Menús interactivos

Aparecen a la derecha de la pantalla y presentan diferentes tipos de información en función del botón seleccionado en el frontal del osciloscopio y de los botones situados a la derecha de la pantalla.

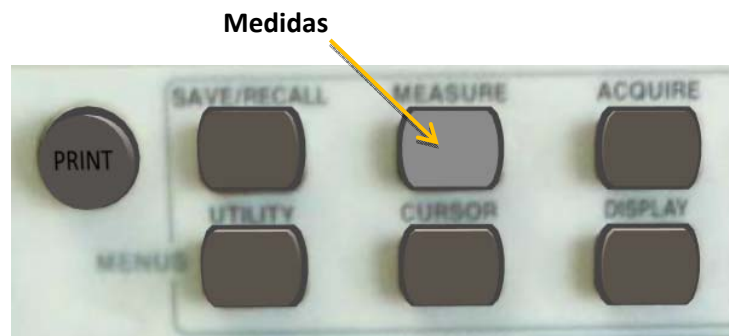
- Menú del tipo de acoplo. En función del canal que se encuentre seleccionado (“CH1 MENU” o “CH2 MENU”) se puede cambiar el tipo de acoplo AC o DC pulsando la tecla situada a la derecha del menú (Figura 5.3-48). En el modo

AC se bloquea la componente continua de la señal que quiera visualizarse, por lo que no se visualiza ni medirá la componente continua de la señal (su valor medio). En el modo DC se visualiza la señal incluyendo la componente continua.



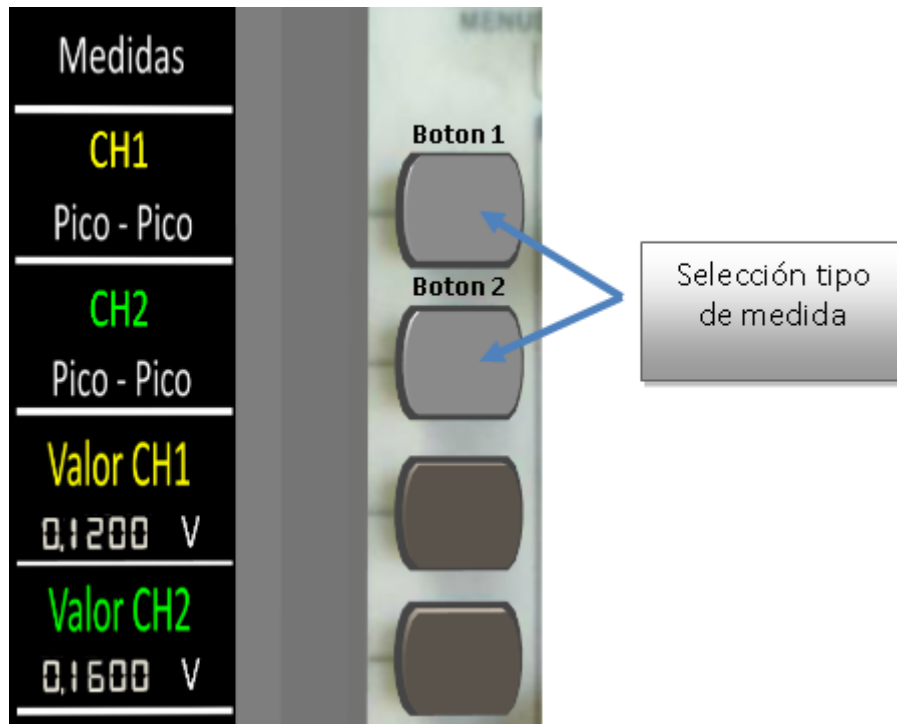
**Figura 5.3-48: Botón para el cambio de acoplamiento**

- Menú de medidas. Al pulsar la tecla MEASURE, situada en la parte superior del osciloscopio, se podrá acceder a los menús de medidas (Figura 5.3-49).

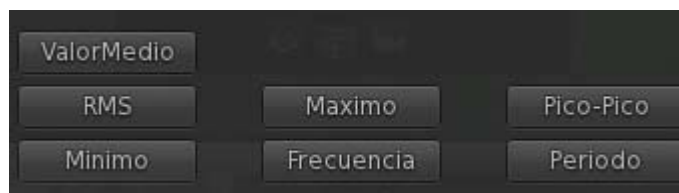


**Figura 5.3-49: Botón para el menú de medidas**

Al pulsar sobre los botones situados a la derecha del menú correspondiente al canal sobre el que se quiera realizar las medidas (Figura 5.3-50) aparecerán dos ventanas donde se podrá elegir el canal y el parámetro de medida (Figura 5.3-51).



**Figura 5.3-50: Botones de selección del tipo de medida**



**Figura 5.3-51: Tipos de medida disponibles**

## Sondas

Para poder conectar las señales a visualizar en el osciloscopio se dispone de dos sondas (Figura 5.3-52), una para el canal 1 y otra para el canal 2. Cada sonda posee, en uno de sus extremos un conector BNC, que se conectará al correspondiente conector de entrada del osciloscopio. En el otro extremo de la sonda existen dos conductores, uno, finalizado en banana, es el conductor de masa que se conectará a la masa de la placa de prueba, y otro, finalizado en punta de prueba, que se conectará al punto de test del circuito que se desee medir. Las sondas de osciloscopio tienen el *factor de atenuación X1* y no se puede modificar.

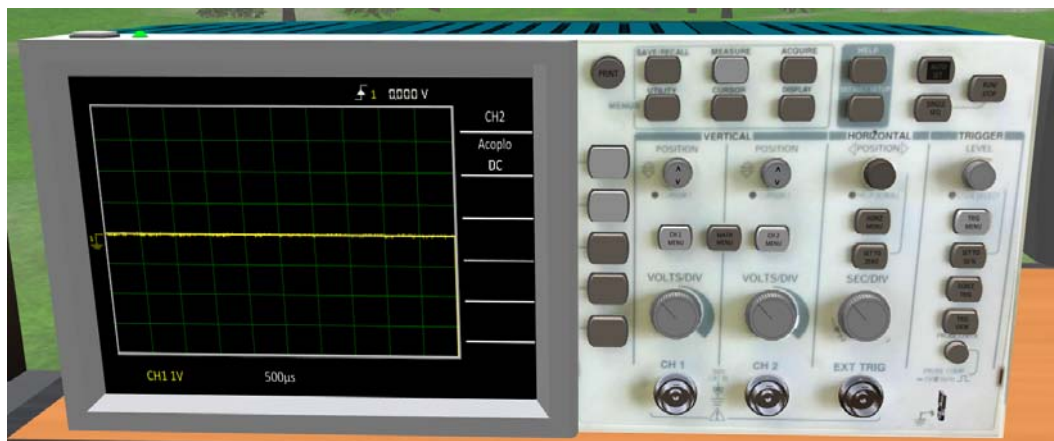




**Figura 5.3-52: Sondas de los canales 1 y 2**

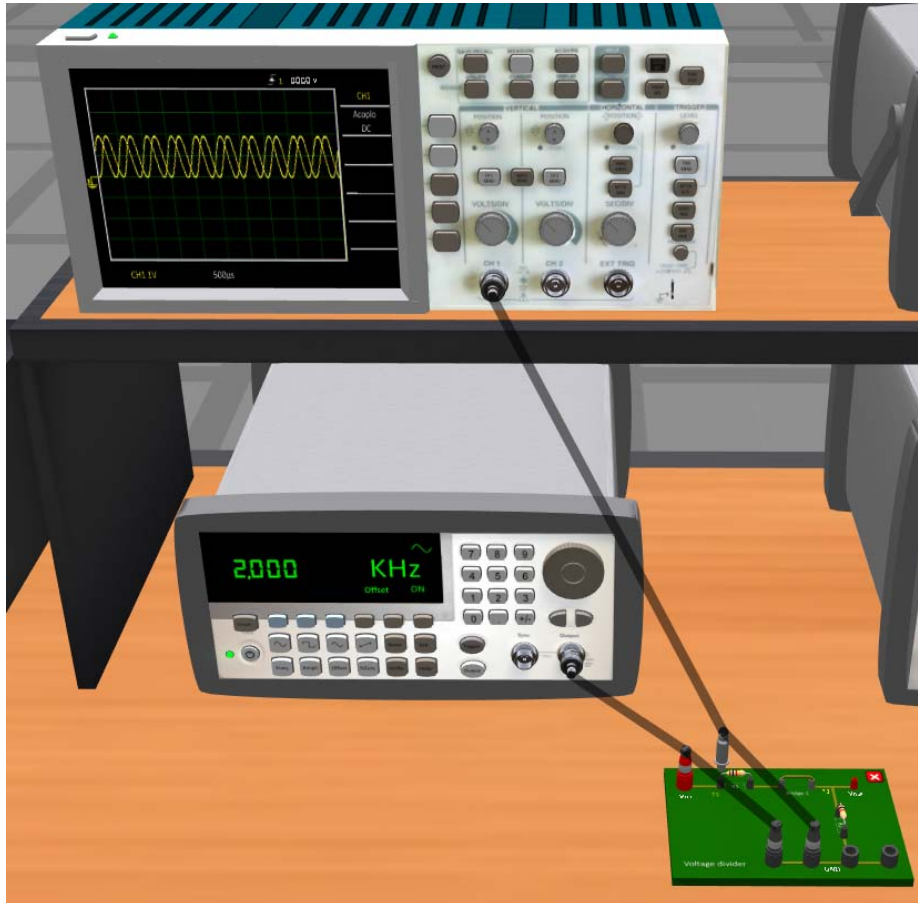
**Acción 31:** encender el osciloscopio pulsando el interruptor situado en la parte superior del instrumento.

**Acción 32:** activar el canal 1 pulsando el botón “CH1 MENU”. En la pantalla del osciloscopio deberá aparecer la traza correspondiente a la señal del canal 1 (Figura 5.3-53).



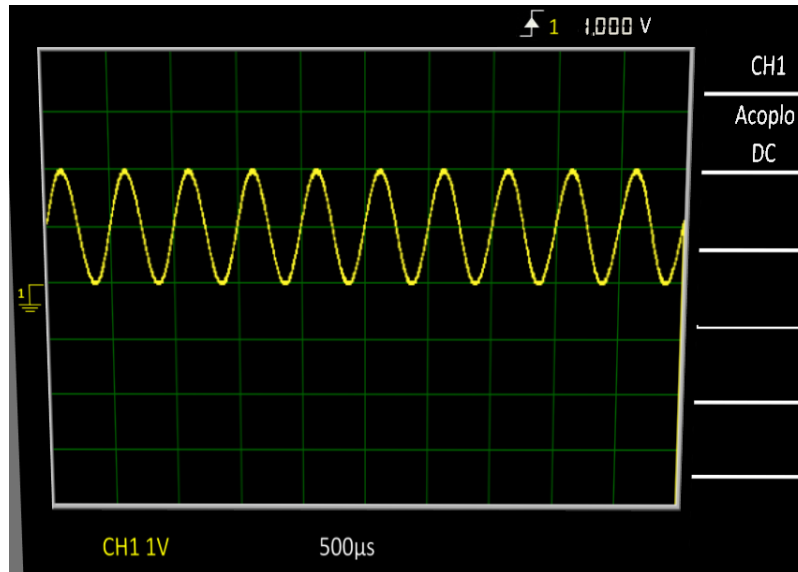
**Figura 5.3-53: Activación del canal 1**

**Acción 33:** conectar la sonda del canal 1 para visualizar la forma de onda que entrega el generador, colocando la punta de prueba de la sonda en la patilla izquierda del resistor R1 (Figura 5.3-54).



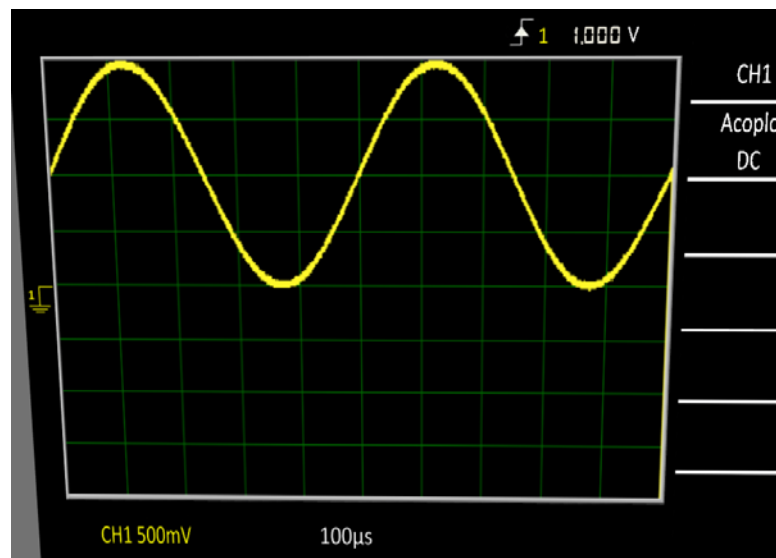
**Figura 5.3-54: Conexión del canal 1 a la placa**

**Acción 34:** configurar el trigger para visualizar correctamente la señal del canal 1. Para ello en primer lugar asegúrese de que el trigger activo es el correspondiente al canal 1 tocando el botón “TRIG MENU”. A continuación varíe el nivel del trigger pulsando el mando giratorio e introduzca un valor de 1V. Observar que la señal visualizada es la que se muestra en la Figura 5.3-55.



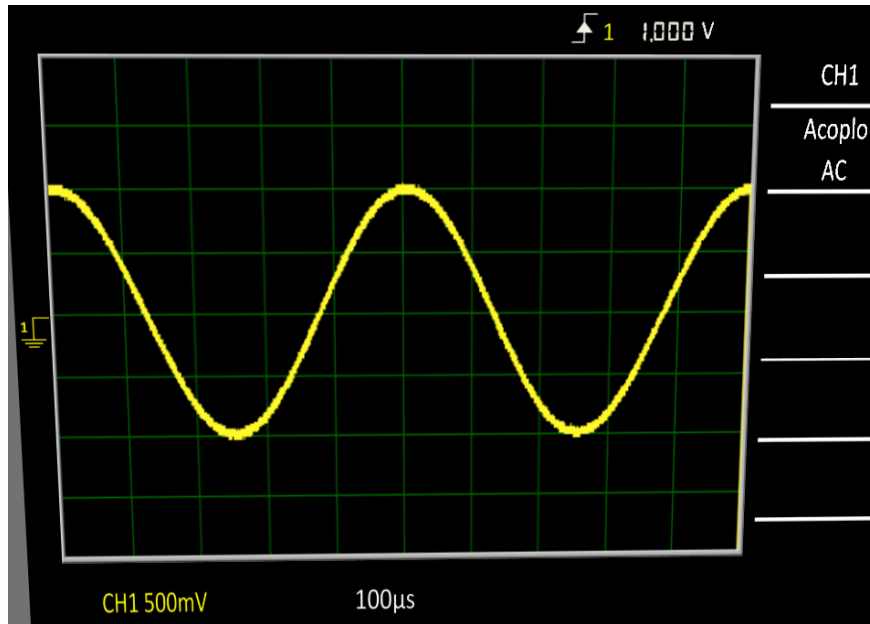
**Figura 5.3-55: Señal a la entrada del circuito**

**Acción 35:** configurar la resolución vertical (V/div) y la base de tiempos, actuando sobre los controles correspondientes, para visualizar correctamente la señal en la pantalla del osciloscopio. Por ejemplo, programar la resolución vertical del canal 1 a 500mV/div y la base de tiempos a 100µs. Observar que la señal visualizada es la que se muestra en la Figura 5.3-56.



**Figura 5.3-56: Señal de entrada con acoplamiento en AC**

**Acción 36:** comprobar la función del acoplamiento AC y DC. Cambiar el modo de acoplamiento a AC. Observar que la señal visualizada aparece sin su componente continua (Figura 5.3-57).



**Figura 5.3-57: Señal de entrada con acoplamiento en DC**

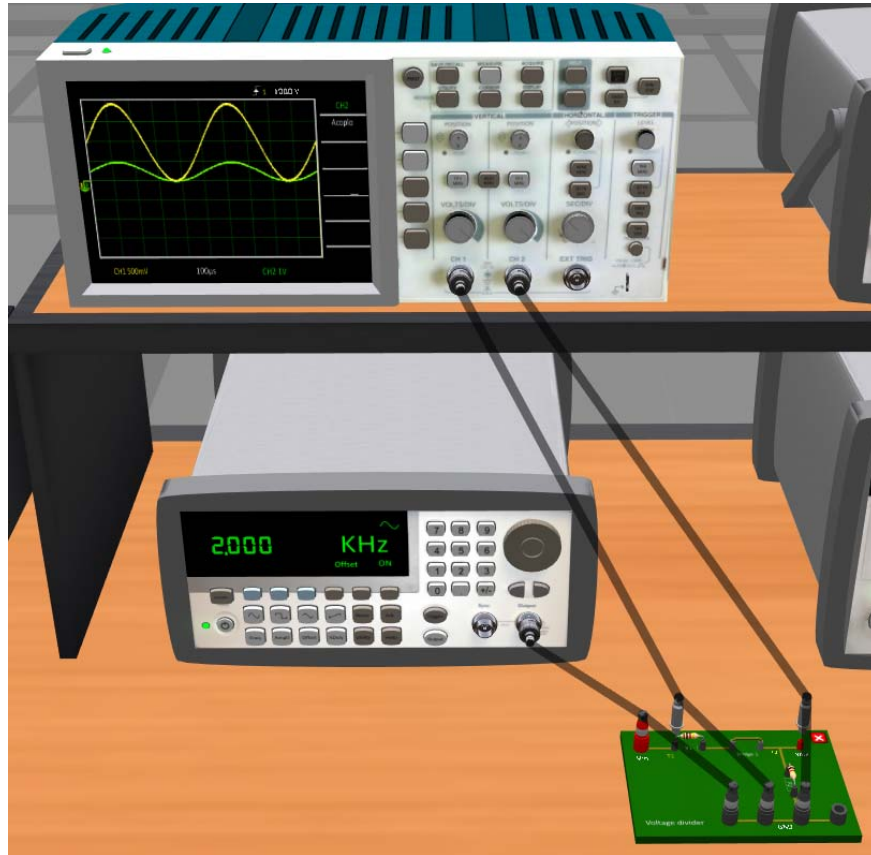
**Acción 37:** Volver a programar el acoplo DC.

**Acción 38:** determinar, de forma teórica, en la placa Divisor de Tensión el valor pico a pico y Valor Medio (componente continua) de la señal que debe existir en Vout. Anotarlo a continuación:

**Valor pico a pico Vout:**

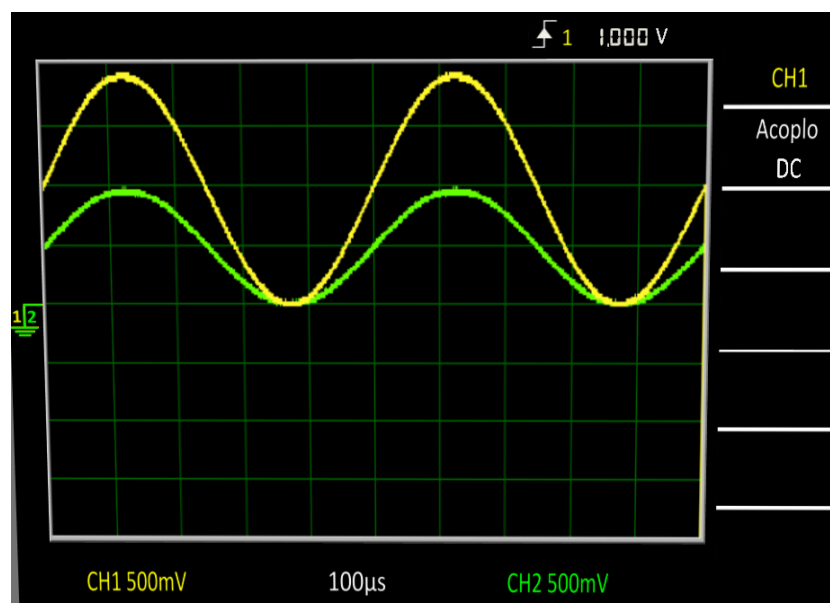
**Valor Medio Vout:**

**Acción 39:** activar el canal 2. Conectar la sonda correspondiente a dicho canal para visualizar la forma de onda de voltaje que hay a la salida del divisor de tensión, colocando la punta de prueba de la sonda en Vout (Figura 5.3-58).



**Figura 5.3-58: Medidas a la entrada y salida del circuito**

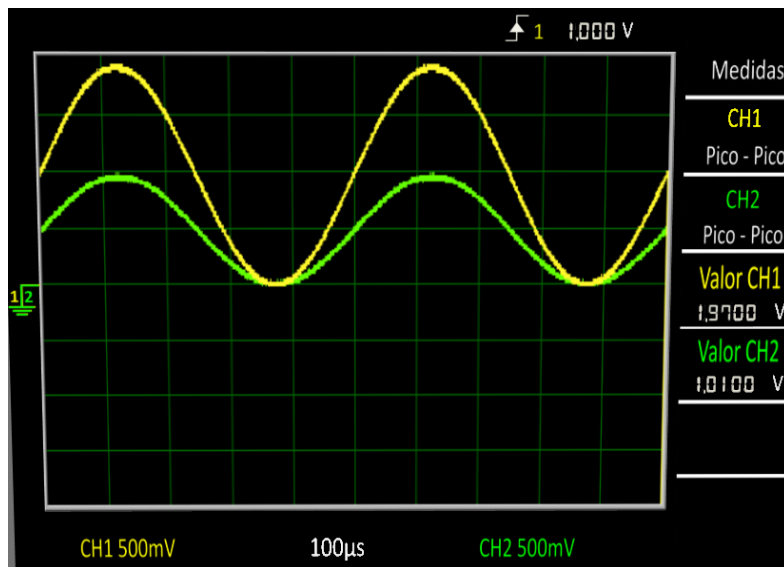
**Acción 40:** configurar la resolución vertical del canal 2 a 500mV/div. Observar que la señal visualizada es la que se muestra en la Figura 5.3-59.



**Figura 5.3-59: Formas de onda obtenidas**

**Acción 41:** realizar la medida del valor pico a pico de la señal Vout mediante el botón “MEASURE”. Anotar el resultado a continuación y observar que la información obtenida es similar a la que se muestra en la Figura 5.3-60.

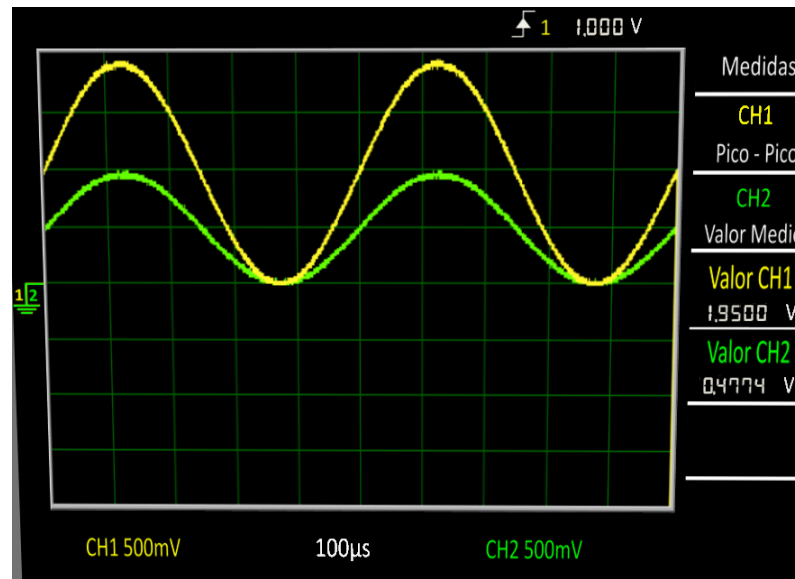
**Valor pico a pico Vout:**



**Figura 5.3-60: Medida de pico-pico a la salida**

**Acción 42:** realizar la medida del Valor Medio de la señal Vout mediante el botón “MEASURE”. Anotar el resultado a continuación y observar que la información obtenida es similar a la que se muestra en la Figura 5.3-61.

**Valor Medio Vout:**



**Figura 5.3-61: Medida del valor medio a la salida**

**Acción 43:** determinar, de forma teórica, en la placa Divisor de Tensión el verdadero valor eficaz del voltaje que debe existir en  $V_{out}$ .

El valor eficaz,  $V_{RMS}$ , del voltaje de una señal alterna con componente continua se calcula utilizando la ecuación:

$$V_{RMS} = \sqrt{V_{RMS(AC)}^2 + V_{DC}^2}$$

Donde  $V_{RMS(AC)}$  es el valor eficaz del voltaje de la señal alterna, eliminada la componente continua, y  $V_{DC}$  es el valor del voltaje de la componente continua.

**Valor eficaz de la señal  $V_{out}$ :**

**Acción 44:** medir el valor eficaz (RMS) de la señal  $V_{out}$  con el osciloscopio. Anotar el resultado a continuación y observar que la información obtenida es similar a la que se muestra en la Figura 5.3-62.

**Valor eficaz de la señal  $V_{out}$ :**

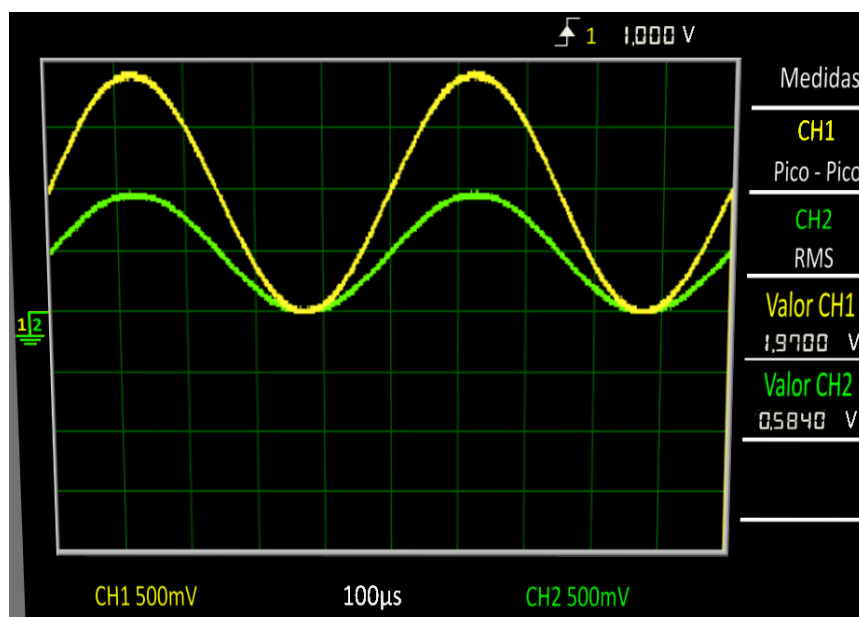


Figura 5.3-62: Medida del valor eficaz la salida

#### 5.3.4.10 Cerrar sesión (logout)

Una vez finalizada la sesión de trabajo en el puesto de laboratorio es necesario recoger todos los elementos utilizados. Se pueden recoger de forma manual de uno en uno o de forma automática al cerrar la sesión.

**Acción 45:** tocar la silla del puesto de laboratorio que ha ocupado y confirmar cerrar la sesión.



### 5.3.5 Pruebas libres

Una vez familiarizado con el uso del laboratorio, tras la realización del tutorial, se encuentra en disposición de poder realizar, de forma libre, algunas acciones adicionales relacionadas con el manejo de los instrumentos y la placa Divisor de Tensión, que pueden permitirle analizar algunas de las posibilidades de aprendizaje que puede proporcionar la plataforma.

A continuación se sugieren posibles acciones a modo de ejemplo:

- Cambiar los resistores de la placa de prueba
- Verificar qué ocurre al quitar algún resistor de la placa
- Modificar valores o parámetros de la señal del generador de funciones que se conecta a la entrada del circuito
- Modificar opciones de visualización y medida de las señales que se muestran en el osciloscopio



# **6**

## **MANUAL DE USUARIO II: SERVIDOR LABORATORIO**

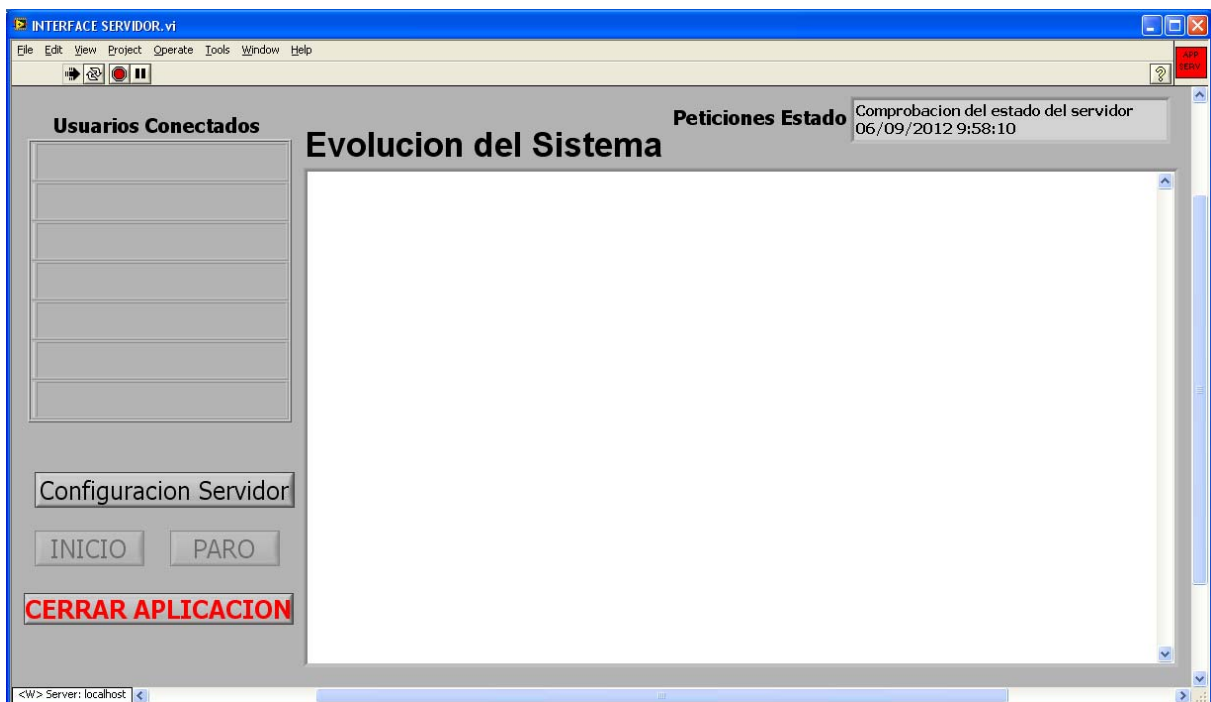


## 6.1 Introducción

Este manual está dirigido al personal encargado del mantenimiento del servidor laboratorio alojado en la Escuela Universitaria de Ingeniería Técnica de Telecomunicación.

El manejo de esta aplicación es muy sencillo, ya que su función principal es mostrar la evolución del sistema y llevar el control de las acciones realizadas por los usuarios a la hora de utilizar el laboratorio de electrónica.

Una vez arrancada la aplicación hay que hacer unas acciones previas para que el servidor laboratorio este en funcionamiento. La ventana que aparece al arrancar es la siguiente (Figura 6.1-1):



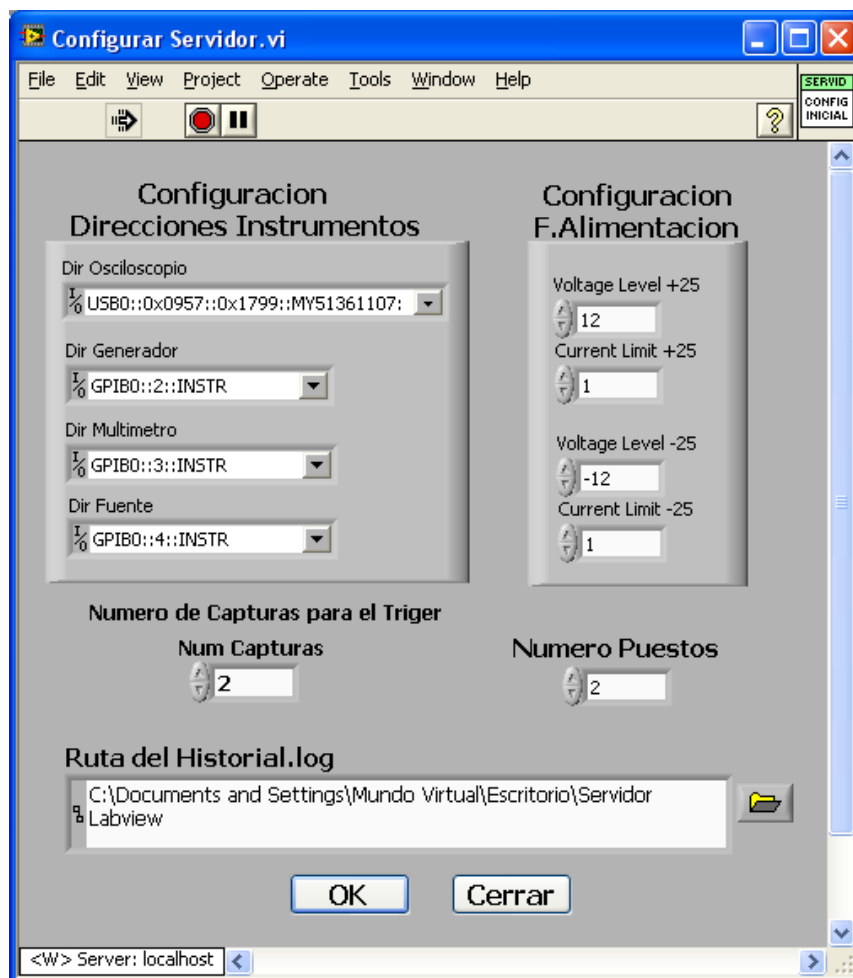
**Figura 6.1-1: Figura 1 Manual Servidor**

Como se puede apreciar, sólo aparecen dos botones disponibles, Configuración Servidor y Cerrar Aplicación. Esto es debido a que antes de que esté en

funcionamiento hay que realizar una configuración previa de algunos aspectos del servidor.

En la parte superior derecha, aparece un cuadro de diálogo que muestra la fecha y hora de la última comprobación, por parte del servidor Opensim, del estado del servidor de laboratorio. Esta comprobación se realiza para indicar a los usuarios que haya en el mundo virtual si es posible o no la interacción con los instrumentos.

**Acción 1:** Hacer clic en el botón *Configurar Servidor*. Aparecerá la ventana de la Figura 6.1-2:



**Figura 6.1-2: Configuración del servidor laboratorio**

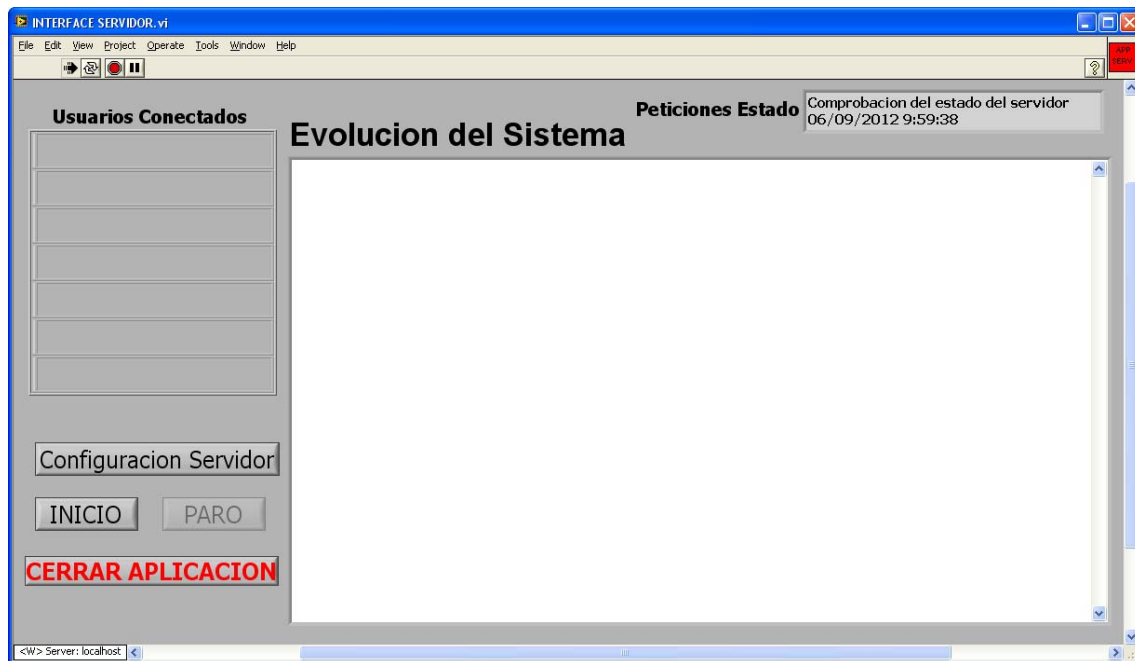
Como se muestra en la figura, hay cinco apartados configurables del servidor:

1. **Configuración de direcciones de Instrumentos:** aquí se escogen las direcciones que tienen asignados los instrumentos según su interface de comunicación.
2. **Configuración de la fuente de alimentación:** se pueden escoger los valores iniciales para el voltaje de salida y limitación de la corriente de la fuente de +25 V. Estos valores se fijan en este momento y no se pueden modificar durante la ejecución de la aplicación.
3. **Número de capturas del trigger:** con este valor definimos cuantas capturas realiza el osciloscopio para ver el efecto del trigger. Cuantas más capturas, más se ralentiza el proceso.
4. **Número de puestos:** debe ser igual al número de puestos operativos que hay en el mundo virtual<sup>12</sup>.
5. **Ruta del fichero Historial.log:** cada vez que se cierra la aplicación, la evolución del sistema que se muestra en la Figura 6.1-1 se guarda en un fichero en la ruta especificada. Si la ruta es siempre la misma, el fichero se va modificando.

**Acción 2:** una vez configurados los parámetros pulsar el botón **OK**. Ahora la ventana que aparece es la siguiente (Figura 6.1-3):

---

<sup>12</sup> Sin tener en cuenta el puesto de laboratorio de la sala de I+D.

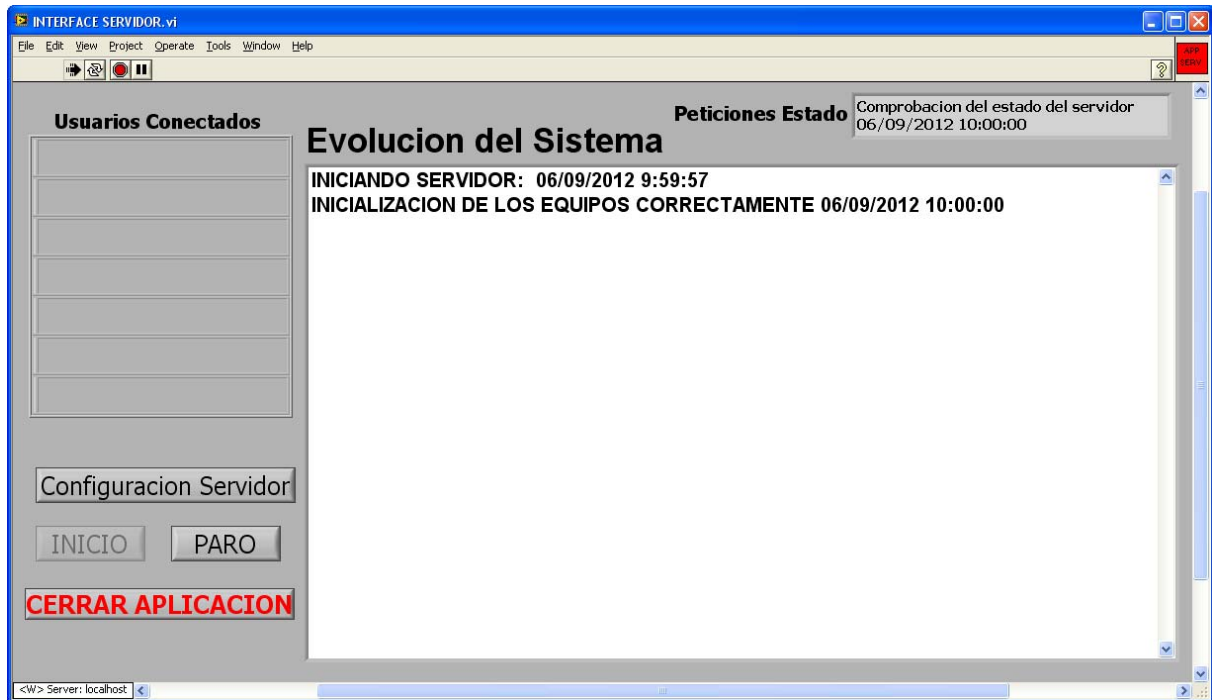


**Figura 6.1-3: Figura 2 Manual Servidor**

Una vez que se ha configurado el servidor y se ha vuelto a la ventana principal, podemos apreciar que ahora está habilitado el botón de **INICIO**.

**Acción 3:** iniciar la aplicación pulsando el botón **INICIO**. Después de unos segundos (que es lo que el servidor tarda en comprobar que todos los instrumentos están conectaos correctamente), como aparece en la Figura 6.1-4, aparece el siguiente mensaje en el indicador **Evolución del Sistema**:





**Figura 6.1-4: Ventana de la aplicación servidor iniciada**

En este momento el servidor del laboratorio estaría iniciado y listo para empezar a atender acciones de usuarios desde el mundo virtual.



# 7 PRESUPUESTO



## 7.1 Software

El software utilizado en este proyecto se puede dividir en dos partes:

- **Software libre:** en esta parte estaría incluido el visor 3D que utilizaría el usuario, el servidor de mundos virtuales Opensim y la aplicación para manejar bases de datos.
- **Software bajo licencia:** aquí estaría el programa LabView de National Instruments. El coste de la licencia para LabVIEW Professional es de **4649 €** (<http://www.ni.com/labview/buy/>). Dicha licencia está disponible para Escuela Universitaria de Ingeniería Técnica de Telecomunicaciones de la UPM, por lo que no conlleva ningún gasto.

En la Tabla 7.1-1 se muestra el coste software:

| CONCEPTO                                      | TOTAL (€)  |
|---|------------|
| <b>TOTAL SOFTWARE<br/>(libre + licencias)</b> | 0 €        |
| <b>TOTAL</b>                                  | <b>0 €</b> |

Tabla 7.1-1: Coste Software

## 7.2 Hardware

Todo el hardware utilizado en el desarrollo del laboratorio remoto, equipos de instrumentación, ordenadores..., pertenece a la Escuela Universitaria de Ingeniería Técnica de Telecomunicaciones de la UPM.

En la Tabla 7.2-1 se muestra el coste hardware:

| CONCEPTO              | TOTAL (€)  |
|-----------------------|------------|
| <b>TOTAL HARDWARE</b> | 0 €        |
| <b>TOTAL</b>          | <b>0 €</b> |

Tabla 7.2-1: Coste Hardware

## 7.3 Mano de obra

Por un lado contamos con las horas de ingeniería libre, y por otro las de redacción del documento. Según el Colegio Oficial de Ingenieros de Telecomunicación (<http://www.coitt.es/res/libredocs/Honorarios.pdf>), el precio de la hora de ingeniería de un libre ejerciente responden al libre acuerdo entre el profesional y su cliente, aunque enumera una serie de conceptos a tener en cuenta para el cálculo. El precio acordado de ingeniería es de 40 €/hora. En cuanto a la realización de la documentación se ha estimado un coste de 16 €/hora. La descripción de horas y coste se observa en la Tabla 7.3-1:

| CONCEPTO  | Nº HORAS | COSTE/HORA (€) | TOTAL (€)       |
|---|----------|----------------|-----------------|
| <b>Especificación del proyecto y búsqueda de soluciones</b> | 50       | 40 €           | 2.000 €         |
| <b>Diseño e implementación Software</b>                     | 500      | 40 €           | 20.000 €        |
| <b>Integración y pruebas</b>                                | 120      | 40 €           | 4.800 €         |
| <b>Documentación y acabado</b>                              | 50       | 16 €           | 800 €           |
| <b>Total ingeniería</b>                                     |          |                | <b>27.600 €</b> |

Tabla 7.3-1: Coste de ingeniería

## 7.4 Total

Para el total del presupuesto de este proyecto, solo hay que tener en cuenta las horas de ingeniería dedicadas, ya que el coste en software y hardware es cero.

| CONCEPTO  | TOTAL (€)       |
|---|-----------------|
| <b>TOTAL SOFTWARE</b><br>(libre + licencias)      | 0 €             |
| <b>TOTAL HARDWARE</b><br>(material + fabricación) | 0 €             |
| <b>TOTAL INGENIERÍA</b>                           | 27.600 €        |
| <b>TOTAL</b>                                      | <b>27.600 €</b> |

**Tabla 7.4-1: Coste total**





# 8

## CONCLUSIONES Y TRABAJO FUTURO



## 8.1 Conclusiones

Se ha desarrollado una aplicación software que, junto con un sistema hardware, crea un laboratorio remoto en el área de la electrónica analógica, siendo una de las partes más atractiva y novedosa de este proyecto la utilización de un **visor 3D** como interface para el usuario final, consiguiendo una mayor inmersión de la realidad.

Los objetivos que se impusieron al principio de este proyecto se han tenido en cuenta para su realización, cumpliendo cada uno de ellos. El sistema final es abierto, modular, versátil y de fácil depuración, por lo que cualquier modificación o futuras ampliaciones se puedan realizar con relativa facilidad y sin suponer un esfuerzo que haga difícil que se lleven a cabo.

En una primera parte, este proyecto me ha servido para conocer en profundidad los entornos virtuales y la gran implantación que pueden llegar a tener en diferentes áreas, pero sobre todo en la docencia, y en concreto en el área de la electrónica analógica.

Otra de las conclusiones que saco de este proyecto es la importancia que tiene el aprendizaje práctico, que a lo largo de la carrera los profesores siempre nos han mencionado, pero no llegamos a darnos cuenta hasta que queremos salir al ámbito laboral.

Uno de los aspectos más positivos que saco de la realización de este proyecto es el trabajo en equipo. Con él se puede conseguir un entorno de trabajo más especializado y versátil, pero siempre apoyándose en los demás compañeros.

## 8.2 Trabajo futuro

Una vez realizado el sistema y cumplidos los objetivos marcados, siempre se podrá mejorar sus funcionalidades e incluso añadir nuevas posibilidades. A su vez, al ser un sistema abierto, cada vez que se diseñe un nuevo módulo de prácticas o se incluya nuevo equipamiento, el sistema aumentará sus posibilidades y opciones.

Las posibles líneas de trabajo futuro que se pueden abordar podrían ser las siguientes:

- Estudiar las posibilidades de integración del laboratorio remoto en los Sistemas de Gestión de Aprendizaje Moodle.
- Incluir más equipos a controlar en el Servidor Laboratorio, pudiendo dar servicio a laboratorios de diferentes áreas.
- Utilizar instrumentos de laboratorio con interfaces de comunicación más rápidos que GPIB.
- Ampliar el número de puestos de laboratorio para dar servicio a un mayor número de usuarios.

9

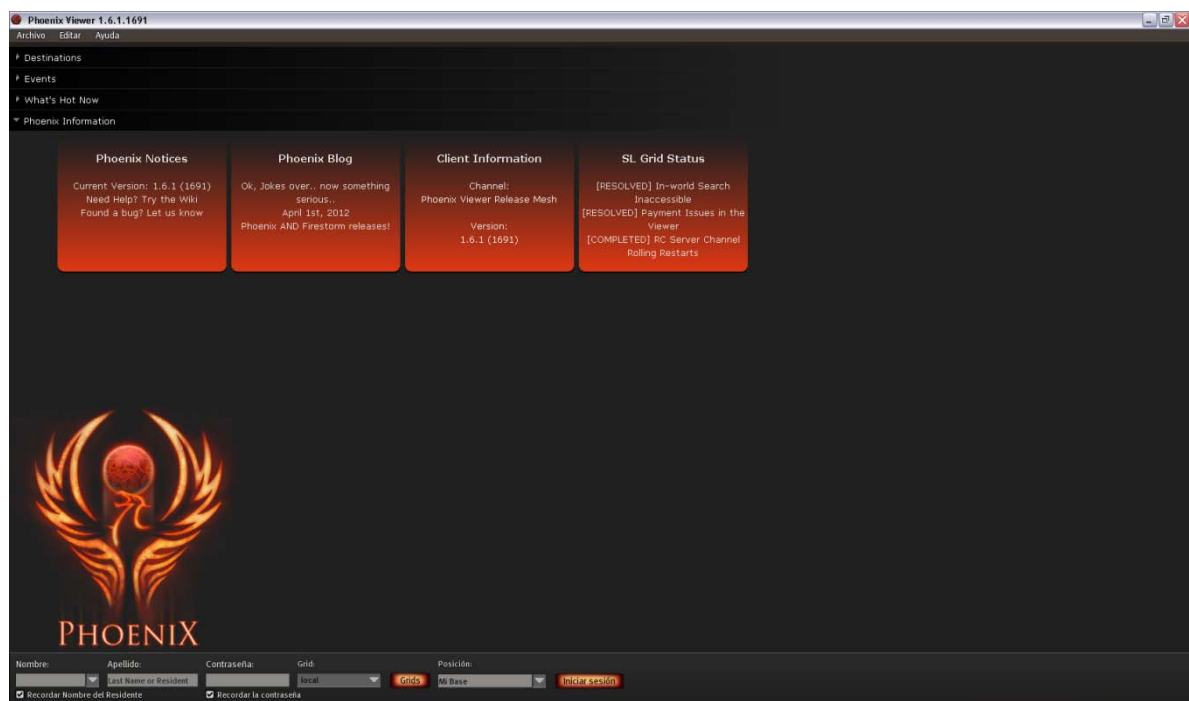
ANEXO I



## 9.1 Instalación Phoenix Viewer

Para instalar el visor Phoenix se deben seguir las siguientes instrucciones:

- Descargar la versión del visor Phoenix (no descargar Firestorm) de la página: <http://www.phoenixviewer.com/downloads.php>
- Instalar el programa aceptando las opciones que aparezcan por defecto. Ejecutar el programa. Deberá aparecer la ventana de la Figura 9.1-1.



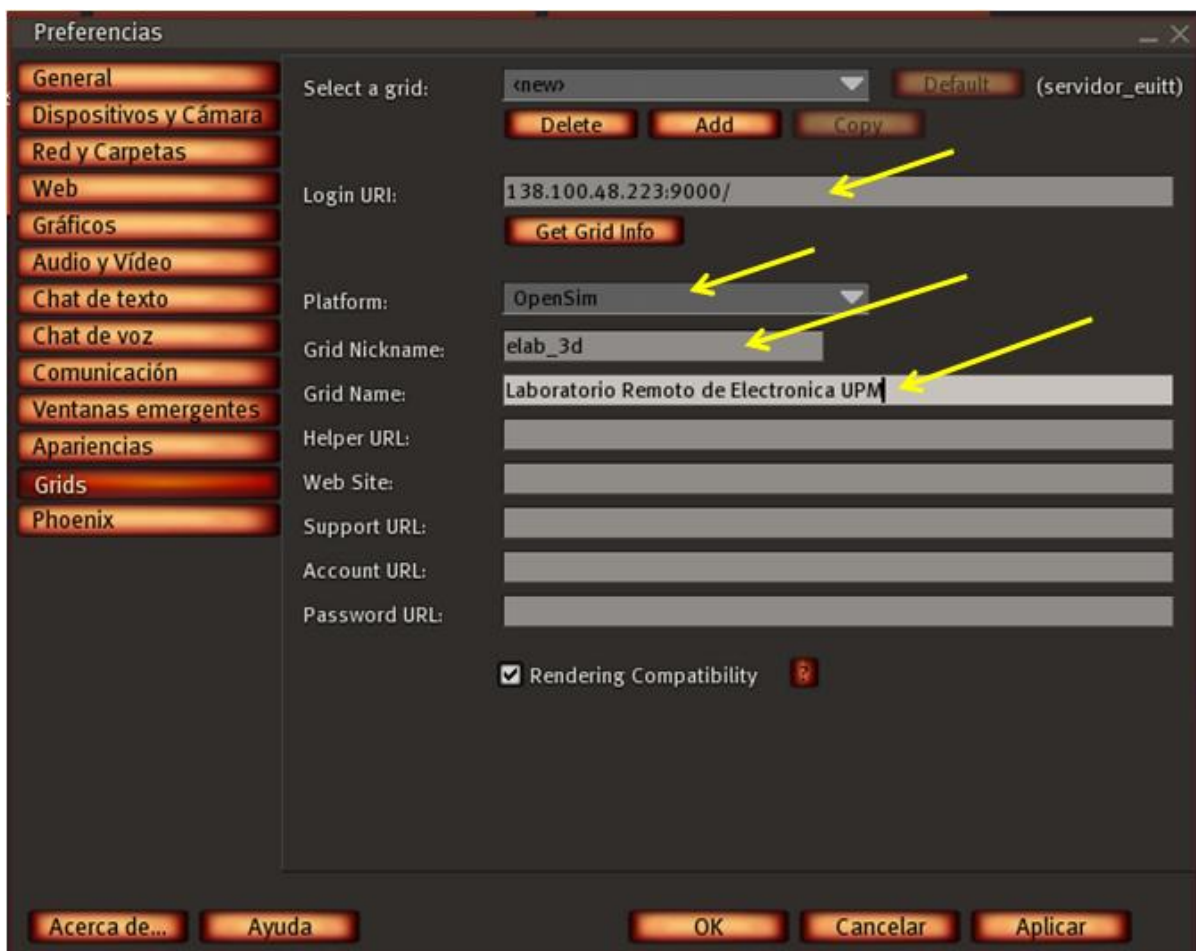
**Figura 9.1-1: Visor Phoenix 1**

- Si los menús aparecen en inglés puede cambiar el idioma accediendo al menú Editar → Preferencias → General → Language. Para que el cambio de idioma tenga efecto deberá cerrar y ejecutar de nuevo el programa.
- En la parte inferior de la pantalla aparecerá los controles que se indican en la Figura 9.1-2.



**Figura 9.1-2: Visor Phoenix 2**

- Para configurar el acceso al Laboratorio de Electrónica de la UPM se debe indicar la dirección donde se encuentra el servidor. Para ello es necesario crear el Grid en el visor. Al hacer clic en el botón “Grids” (Figura 9.1-2), aparecerá una ventana en la que deberá pulsar el botón “Add”, debiendo configurar los campos indicados en la Figura 9.1-3.

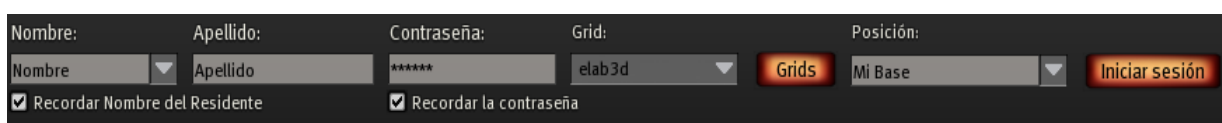


**Figura 9.1-3: Visor Phoenix 3**

Salir de esta ventana de configuración haciendo clic en Aplicar y luego en OK. Al cerrarse la ventana se volverá a la pantalla inicial (Figura 9.1-1).



- Realizar las siguientes acciones teniendo en cuenta los controles que se muestran en la Figura 9.1-4.
  - Introducir el nombre de usuario y la contraseña que se le haya asignado
  - En el desplegable Grid seleccionar “elab\_3d”
  - En “Posicion” seleccione “Mi base”
  - Hacer clic en Iniciar Sesión

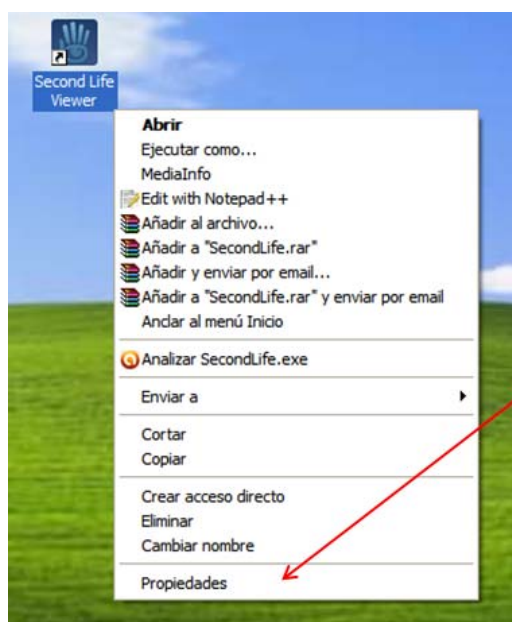


**Figura 9.1-4: Visor Phoenix 4**

## 9.2 Instalación SecondLife Viewer

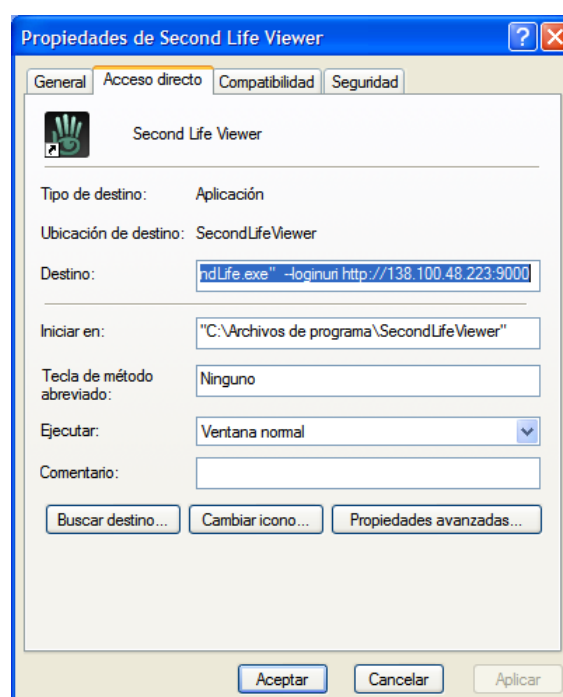
Para instalar el visor SecondLife se deben seguir las siguientes instrucciones:

- Descargar la última versión del visor SecondLife de la página: <http://secondlife.com/support/downloads/>
- Instalar el programa aceptando las opciones que aparezcan por defecto.
- Antes de ejecutar el programa deberá configurar el acceso al Laboratorio de Electrónica de la UPM. Para ello debe hacer clic con el botón derecho sobre el acceso directo del visor SecondLife en el escritorio y seleccionar el menú de Propiedades (Figura 9.2-1).



**Figura 9.2-1: Visor SecondLife 1**

- En el campo Destino se sustituye el final del recuadro “--set InstallLanguage es” por -loginuri <http://138.100.48.223:9000/>. Como se muestra en la Figura 9.2-2.



**Figura 9.2-2: Visor SecondLife 2**

- Ejecutar el programa.
- Introducir los datos de usuario e iniciar sesión.

# 10

## REFERENCIAS



## 10.1 Documentos

- LÓPEZ GREGORIO, Sergio. Estudio y definición de un modelo de arquitectura hardware y software para la implementación de un laboratorio remoto en el área de la electrónica analógica. Trabajo de investigación. Universidad Politécnica de Madrid, Madrid Julio 2010.
- Cursos LabVIEW Intermediate I y II. National Instruments.
- DEL ALAMO, J. A., et al. MIT Microelectronics Weblab. En: FJELDLY, T. A. y SHUR, M. S. (eds.) Lab on the web. Running real electronics experiments via the Internet. John Wiley & Sons, 2003. Capítulo 2, pp. 49-88.
- FJELDLY, T. A. y SHUR, M. S. (eds.). Lab on the Web. Running Real Electronics Experiments Via the Internet. John Wiley & Sons, 2003. ISBN 0-471-41375-5.
- GARCIA-ZUBIA, J., et al. Addressing Software Impact in the Design of Remote Laboratories. IEEE Transactions on Industrial Electronics, 2009b, vol. 56, no. 12, pp. 4757-4767. ISSN 0278-0046.
- GARCIA-ZUBIA, J.; LOPEZ-DE-IPINA, D. y ORDUNA, P. Evolving Towards Better Architectures for Remote Laboratories: A Practical Case. International Journal of Online Engineering, 2009a, vol. 1, no. 2 Spec. Iss. ISSN 1861-2121.
- GARCIA-ZUBIA, J., et al. Remote Laboratories from the Software Point of View. En: GOMES, L. y GARCIA\_ZUBIA, J. G (eds.). Advances on Remote Laboratories and e-Learning Experiences. Universidad de Deusto, 2007. Capítulo 6, pp. 131-150.
- SAN CRISTOBAL RUIZ, Elio. Metodología, estructura y desarrollo de interfaces intermedias para la conexión de laboratorios remotos y virtuales a plataformas educativas. Tesis Doctoral, 2010.
- CRUZ RUIZ, Fernando; DURBEY CARRASCO, Alberto; SANZ BRIONES, Jorge. Entorno virtual 3D multiusuario para la simulación de escenarios de evacuación. Proyecto de Sistemas Informáticos Facultad de Informática de la Universidad Complutense de Madrid, Madrid 2010-2011.

## 10.2 Consultas on-line

- Web oficial de bus USB: <http://www.usb.org>
- Wikipedia: <http://es.wikipedia.org/wiki/USB>
- Web National Instruments: <http://www.ni.com>
- Crear servicio web labview: [http://zone.ni.com/reference/en-XX/help/371361G-01/lvhowto/build\\_web\\_service/](http://zone.ni.com/reference/en-XX/help/371361G-01/lvhowto/build_web_service/)
- LabView y GPIB: [www.ni.com/gpib](http://www.ni.com/gpib)
- Página de OpenSimulator: [http://opensimulator.org/wiki/Main\\_Page](http://opensimulator.org/wiki/Main_Page)
- Página principal de OsGrid: <http://www.osgrid.org/>
- FAQ de OpenSim:
  - <http://opensimulator.org/wiki/EspriFAQ>
  - <http://opensimulator.org/wiki/Configuration>
  - [http://opensimulator.org/wiki/Configuring\\_Regions](http://opensimulator.org/wiki/Configuring_Regions)
- Página principal de OsGrid: <http://www.osgrid.org/>
- Openlife: <http://openlifegrid.com/>
- Foro LibOpenMetaverse: <http://libsecondlife.13613.n2.nabble.com/>
- Wiki Second Life: [http://wiki.secondlife.com/wiki/Main\\_Page](http://wiki.secondlife.com/wiki/Main_Page)
- SecondLife: <http://secondlife.com/support/downloads/>
- LSL:
  - [http://wiki.secondlife.com/wiki/LSL\\_Portal](http://wiki.secondlife.com/wiki/LSL_Portal)
  - <http://isgametime.wordpress.com/category/tutorial-lsl/>
- Visores Firestorm y Phoenix:  
<http://www.phoenixviewer.com/downloads.php>
- Requerimientos software y hardware para Second Life:  
<http://secondlife.com/support/system-requirements/>
- Hispagrid: <http://foro.hispagrid.com/viewforum.php?f=5>
- Irene Muni: <http://www.irenemuni.com/foro/opensim-vf14.html>
- Atlante Guerrero: <http://www.creasl.org/>
- New World: <http://www.newworldgrid.com/lang/en-us/>
- Agilent: [www.agilent.com](http://www.agilent.com)
- Hewlett Packard: [www.hp.com](http://www.hp.com)

- Tektronix: [www.tektronix.com](http://www.tektronix.com)
- Agilent 33120A:
  - Data sheet: <http://www.datasheetcatalog.org/datasheet2/d/0jrc7i171udhujz709zrrcsq95py.pdf>
  - User guide: <http://mntl.illinois.edu/equipment/docs/agilent33120auserguide.pdf>
- Hewlett Packard 34401A
  - Data sheet: [http://www.metrictest.com/catalog/pdfs/product\\_pdfs/hp\\_34401a.pdf](http://www.metrictest.com/catalog/pdfs/product_pdfs/hp_34401a.pdf)
  - User guide: <http://www.ee.buffalo.edu/courses/elab/hp34401.pdf>
- Hewlett Packard E3631A
  - Data sheet: <http://cp.literature.agilent.com/litweb/pdf/E3631-90002.pdf>
  - User guide: [http://www.equipland.com/objects/catalog/product/extras/10386\\_E363xA.pdf](http://www.equipland.com/objects/catalog/product/extras/10386_E363xA.pdf)
- Agilent DSO-X 2012A
  - Data sheet: <http://cp.literature.agilent.com/litweb/pdf/5990-6619ESE.pdf>
  - User Guide: <http://cp.literature.agilent.com/litweb/pdf/75015-97033.pdf>
- VISIR (Virtual Instrument System in Reality, Blekinge Institute of Technology, Suecia): <https://weblab-visir.deusto.es/electronics/>
- ELVIS iLab, Massachusetts Institute of Technology (MIT, USA): <http://openilabs.mit.edu/ServiceBroker/>
- ISILab (Internet shared Instrumentation Laboratory, Universidad de Génova (Italia): <http://isilab.dibe.unige.it/>
- NetLab, Universidad de South Australia (UniSA): <http://netlab.unisa.edu.au/index.xhtml>

